wikiHow to do anything...

How to Create a Simple Web Page with HTML

PARTS

OTHER SECTIONS

Written by Nicole Levine, MFA

Last Updated: February 2, 2024 Fact Checked

This wikiHow teaches you how to write a simple web page with HTML (hypertext markup language). HTML is one of the core components of the World Wide Web, making up the structure of web pages. Once you've created your web page, you can save it as an HTML document and view it in your web browser. Creating an HTML page is possible using basic text editors found on both Windows and Mac computers.

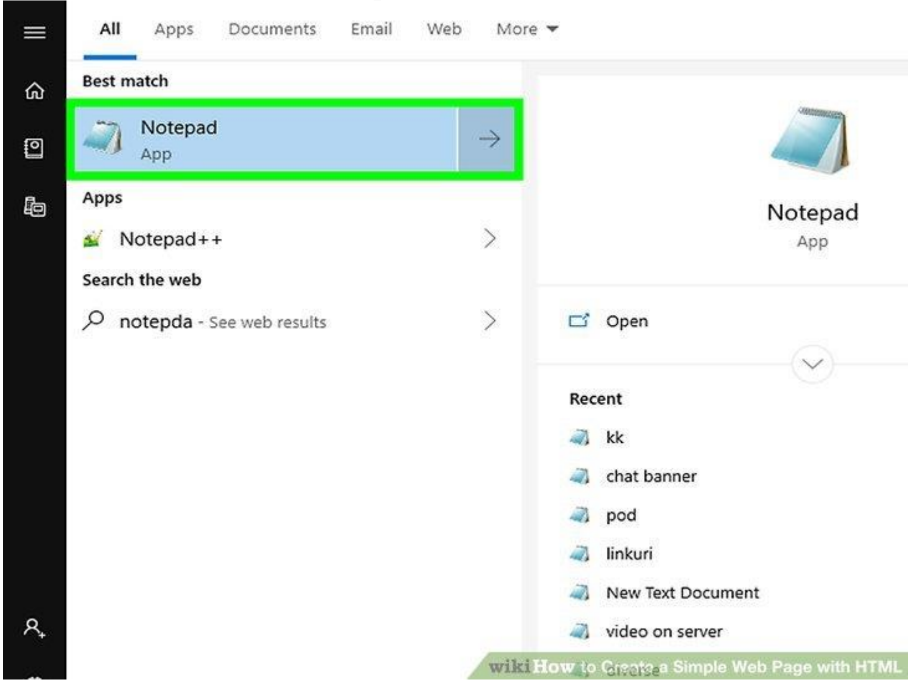
Part 1

Adding a Head to Your HTML

1

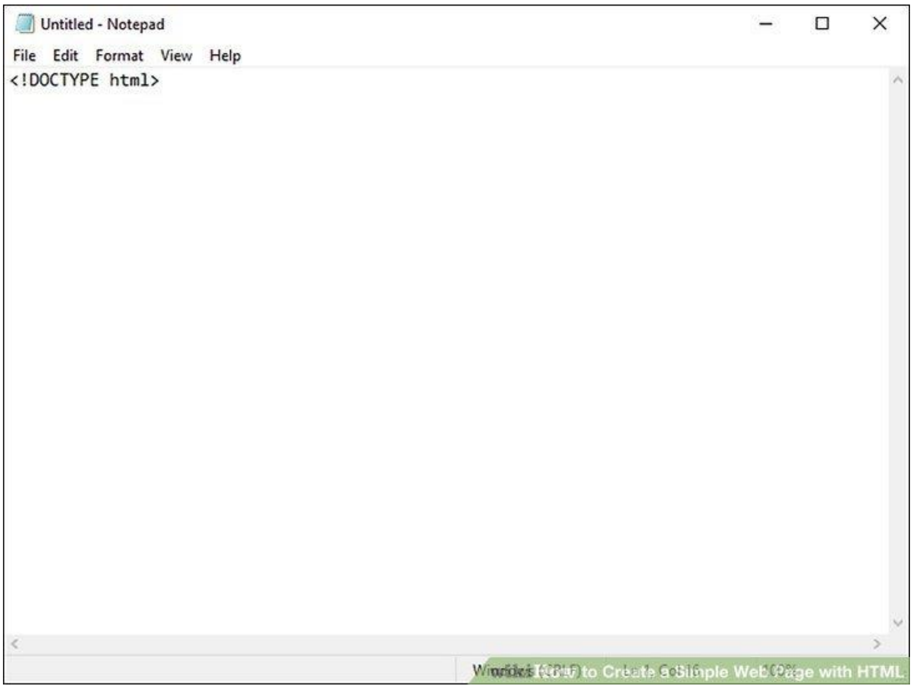
Open a text editor. On a computer running the Windows operating system, you'll usually use Notepad, or Notepad++ whereas macOS users will use TextEdit and ChromeOS users will use Text:

- *Windows* - Open **Start** , type in `notepad`, or `notepad++` and click **Notepad** or "Notepad++ or sublime" at the top of the window.
- *macOS* - Click **Spotlight** , type in `textedit`, and double-click **TextEdit** at the top of the results.
- *ChromeOS* - Open launcher, then click Text. (The icon says Code Pad).



2

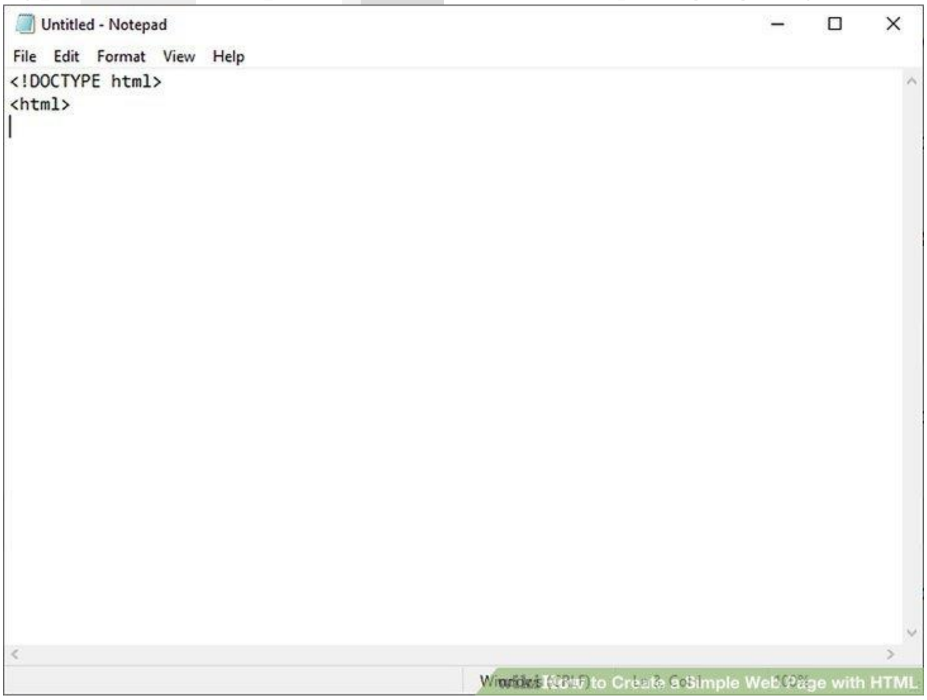
Type in `<!DOCTYPE html>` and press ↵ Enter. This tells the web browser that this is an HTML document.[1]


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple Web Page with HTML

```
<!DOCTYPE html>
```

3

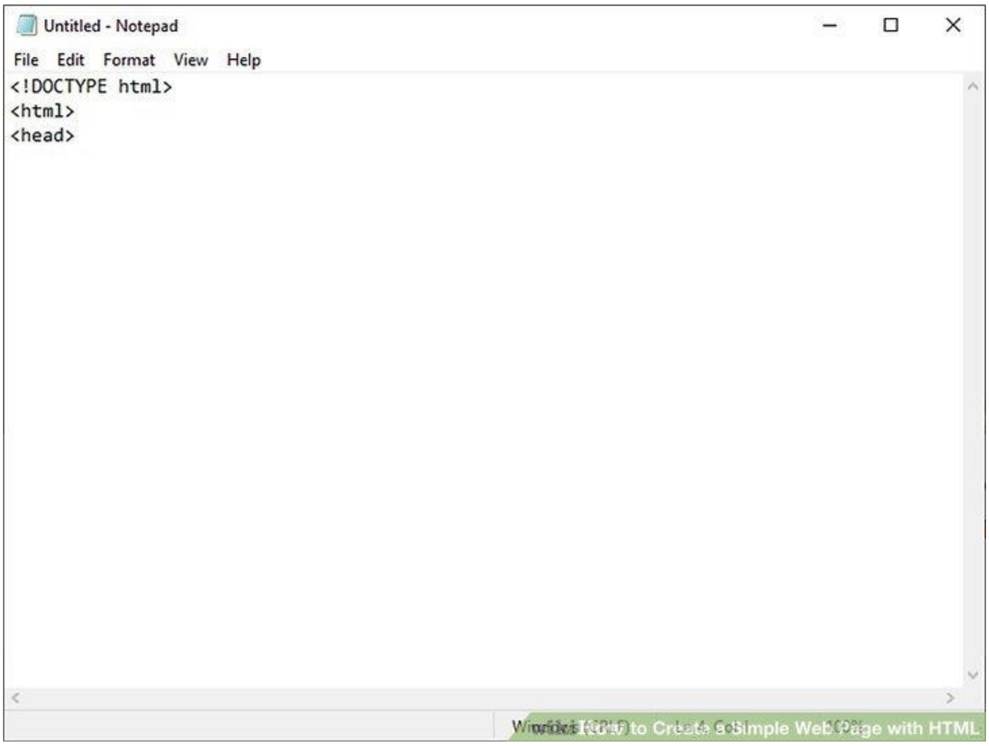
Type `<html>` and press ↵ Enter. This is the opening tag for your HTML code.


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple Web Page with HTML

```
<!DOCTYPE html>
<html>
```

4

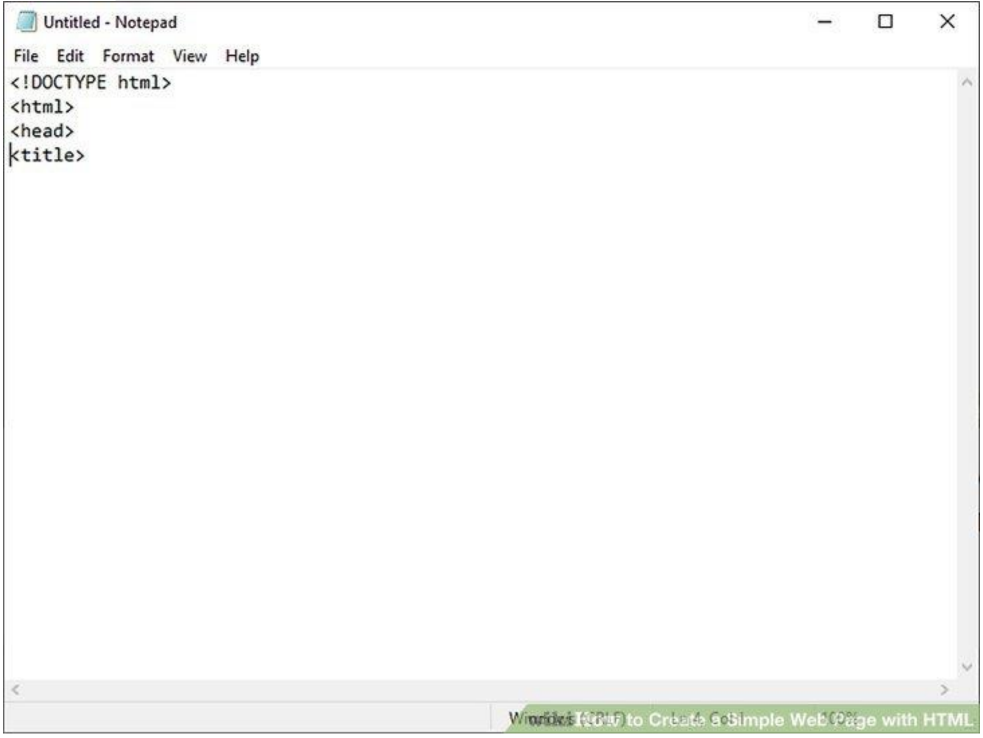
Type in `<head>` and press ↵ Enter. This is the tag that opens your HTML head. The HTML head information that is not usually displayed on your web page. This information can include, the title, meta data, CSS style sheets (Cascading Style Sheet), and other scripting languages.[2]


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple Web Page with HTML

```
<!DOCTYPE html>
<html>
<head>
```

5

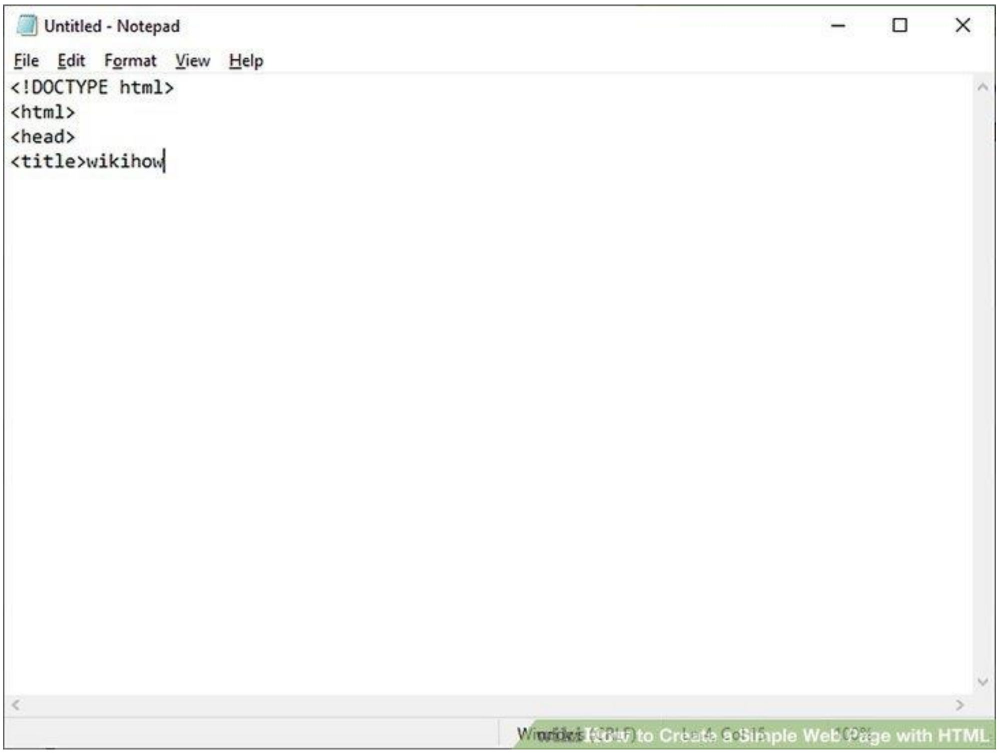
Type in `<title>`. This is the tag to add a title to your page.[3]


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple Web Page with HTML

```
<!DOCTYPE html>
<html>
<head>
<title>
```

6

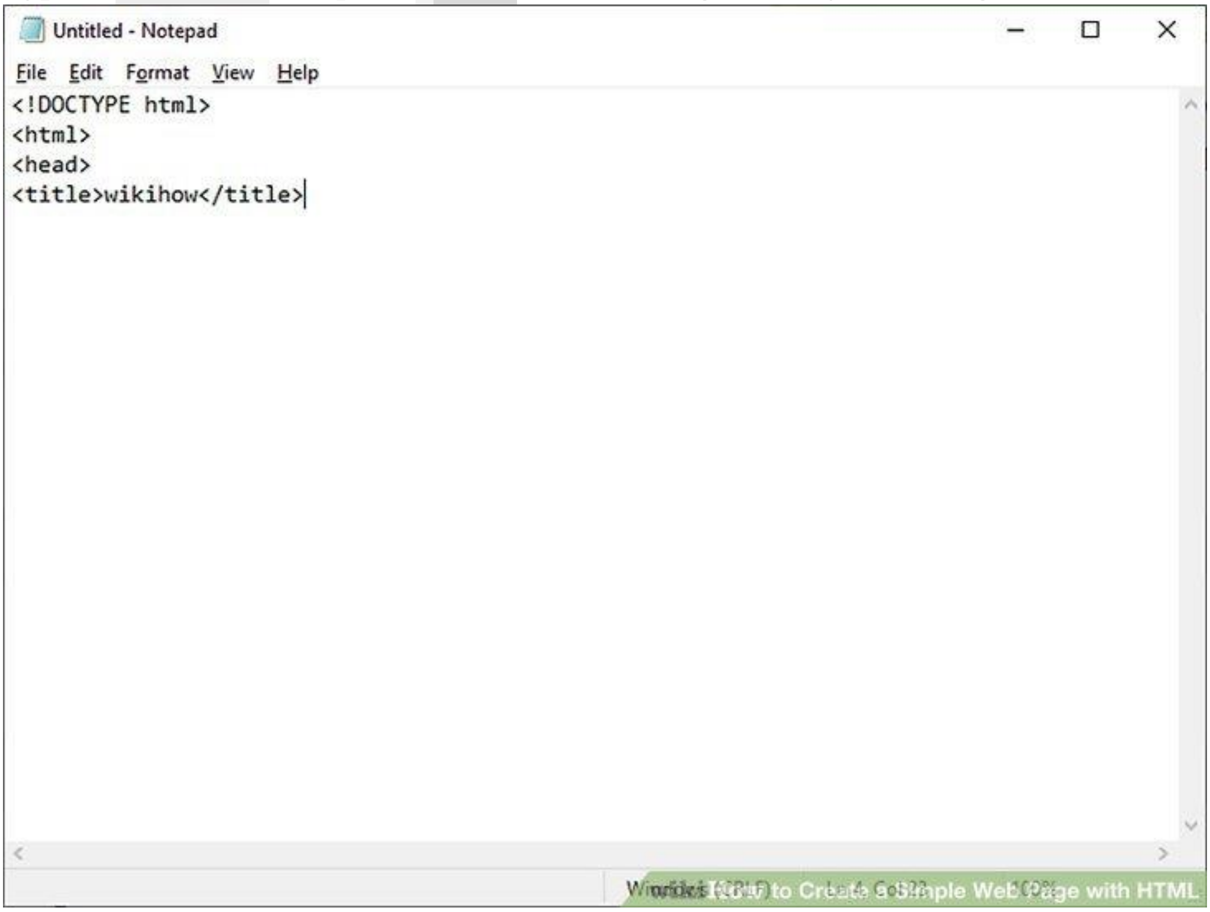
Type a title for your web page. This can be any title you want to name your web page.


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple Web Page with HTML

```
<!DOCTYPE html>
<html>
<head>
<title>wikihow
```

7

Type in `</title>` and press ↲ Enter. This is the tag to close your title tag.

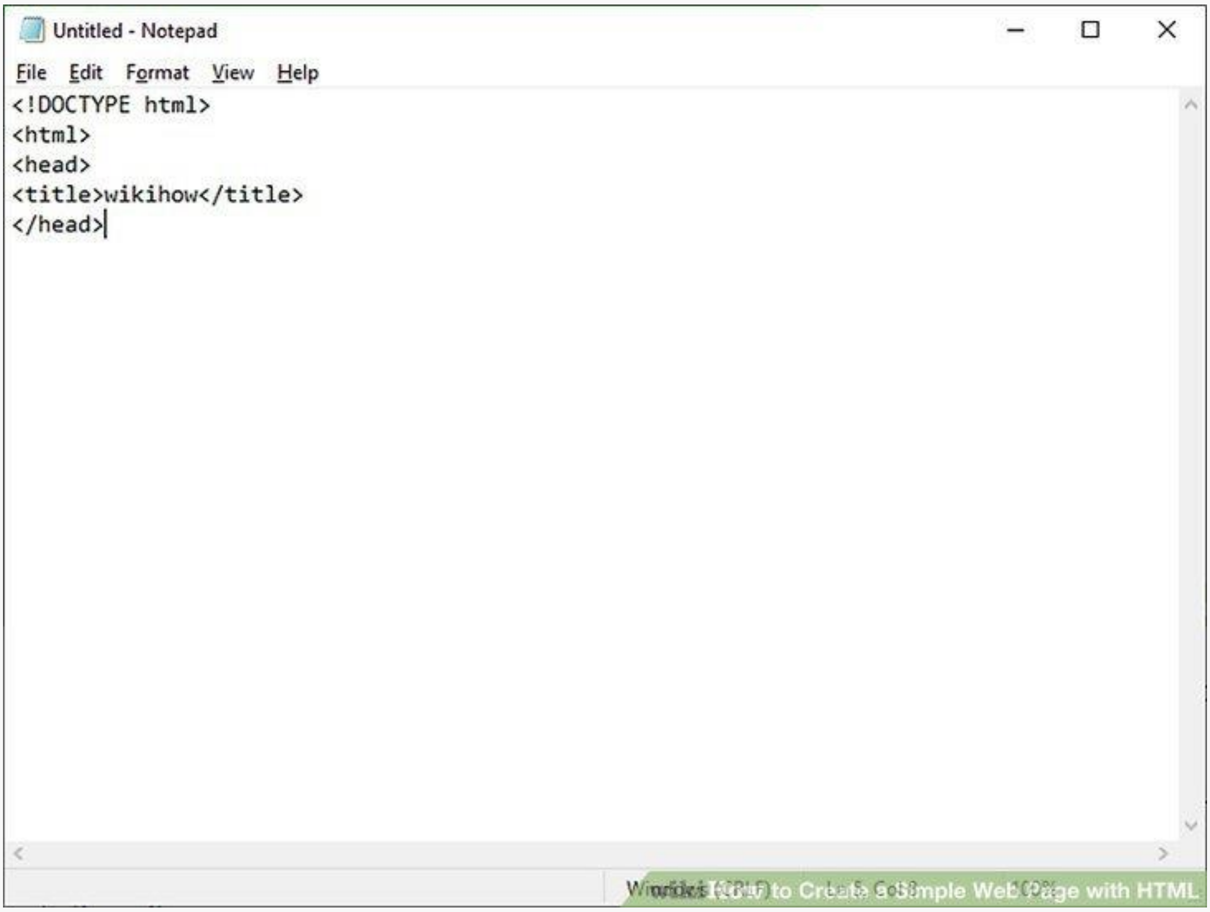

Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple Web Page with HTML

```
<!DOCTYPE html>
<html>
<head>
<title>wikihow</title>
```

8

Type `</head>` and press ↵ Enter. This is the tag to close your head. Your HTML code should look something like this.

```
<!DOCTYPE html>
<html>
<head>
<title>My Web Page</title>
</head>
```


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple Web Page with HTML

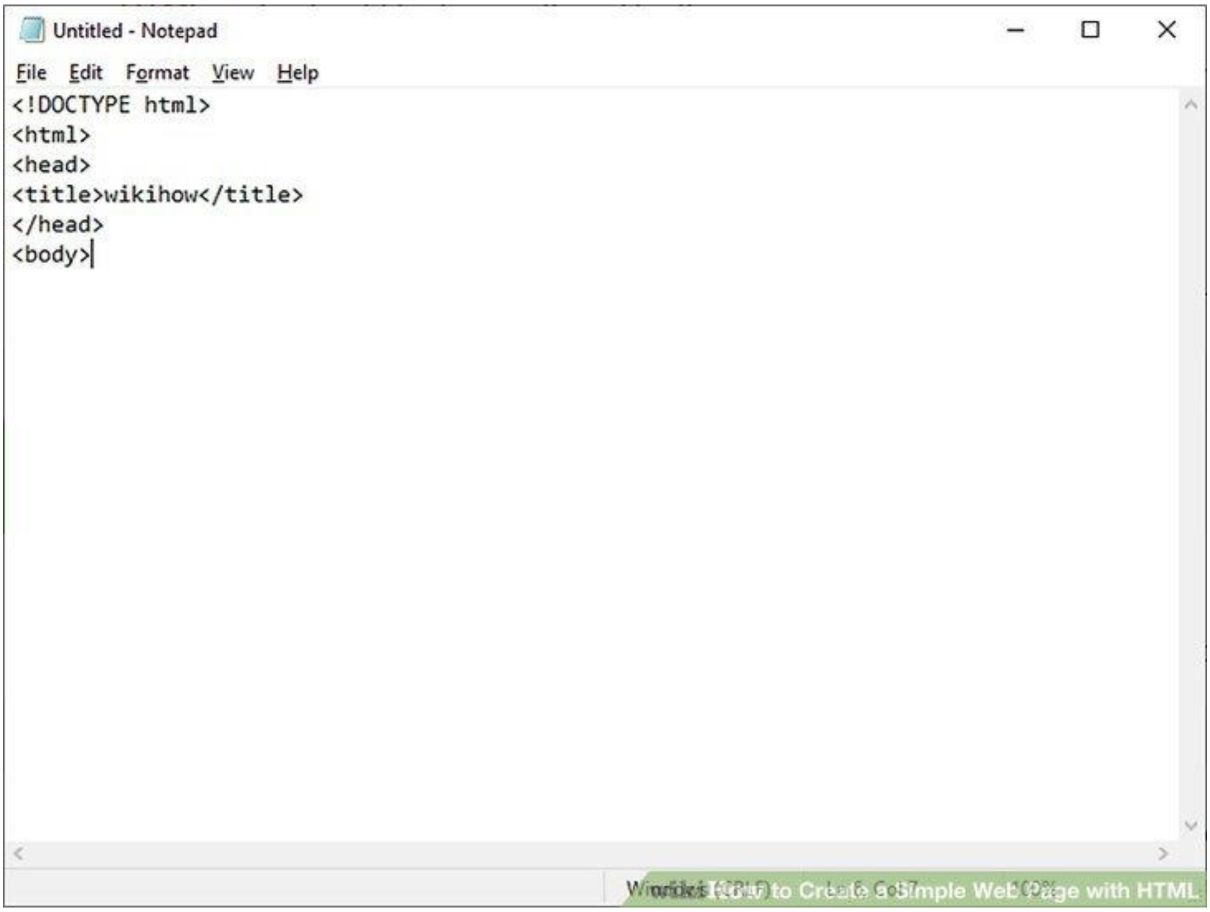
```
<!DOCTYPE html>
<html>
<head>
<title>wikihow</title>
</head>
```

Part 2

Adding a Body and Text to Your HTML

1

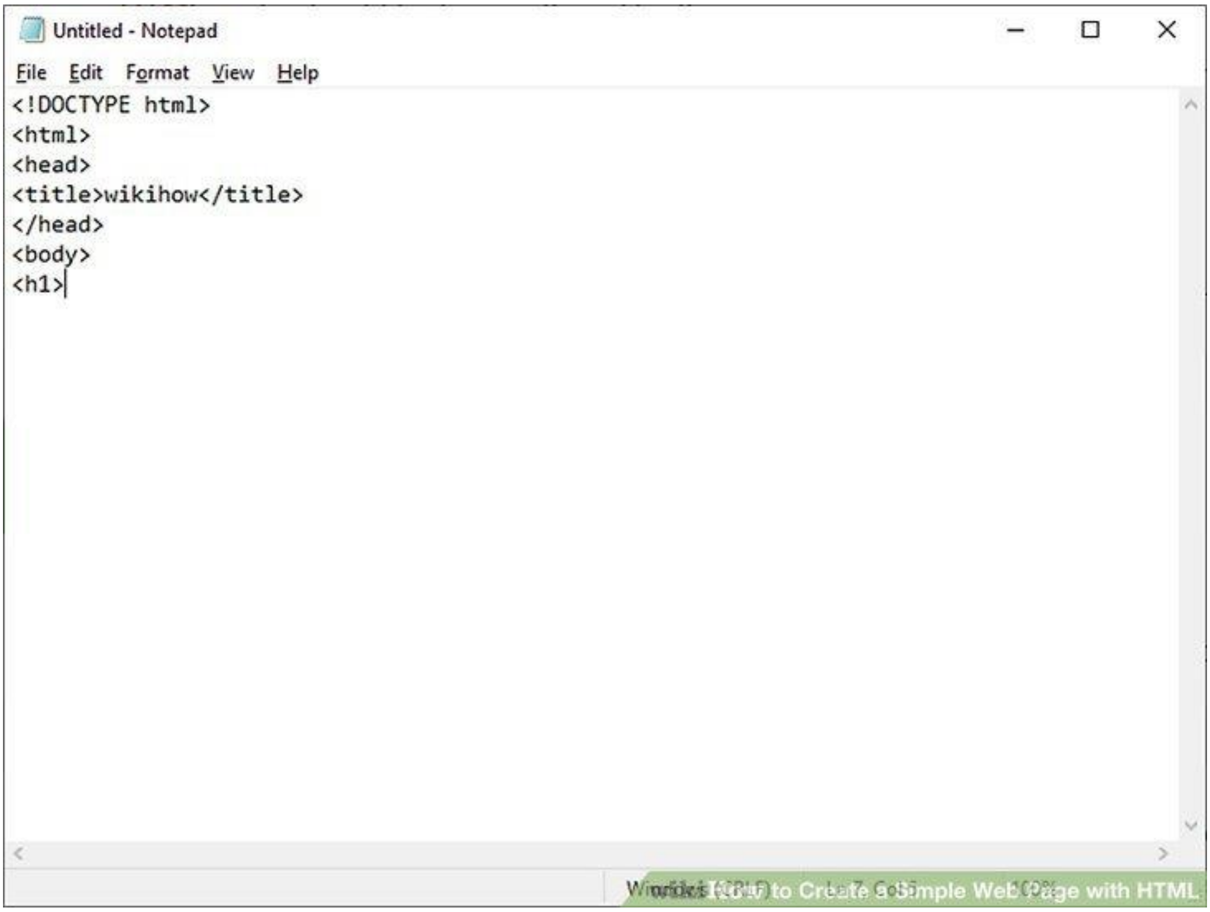
Type in `<body>` below the closed "Head" tag. This tag opens the body of your HTML document. Everything that goes in the HTML body displays on the web page.[4]



```
<!DOCTYPE html>
<html>
<head>
<title>wikihow</title>
</head>
<body>
```

2

Type in `<h1>`. This is the tag to add a heading to your HTML document. A Heading is large bold text that typically goes at the top of your HTML document.[5]

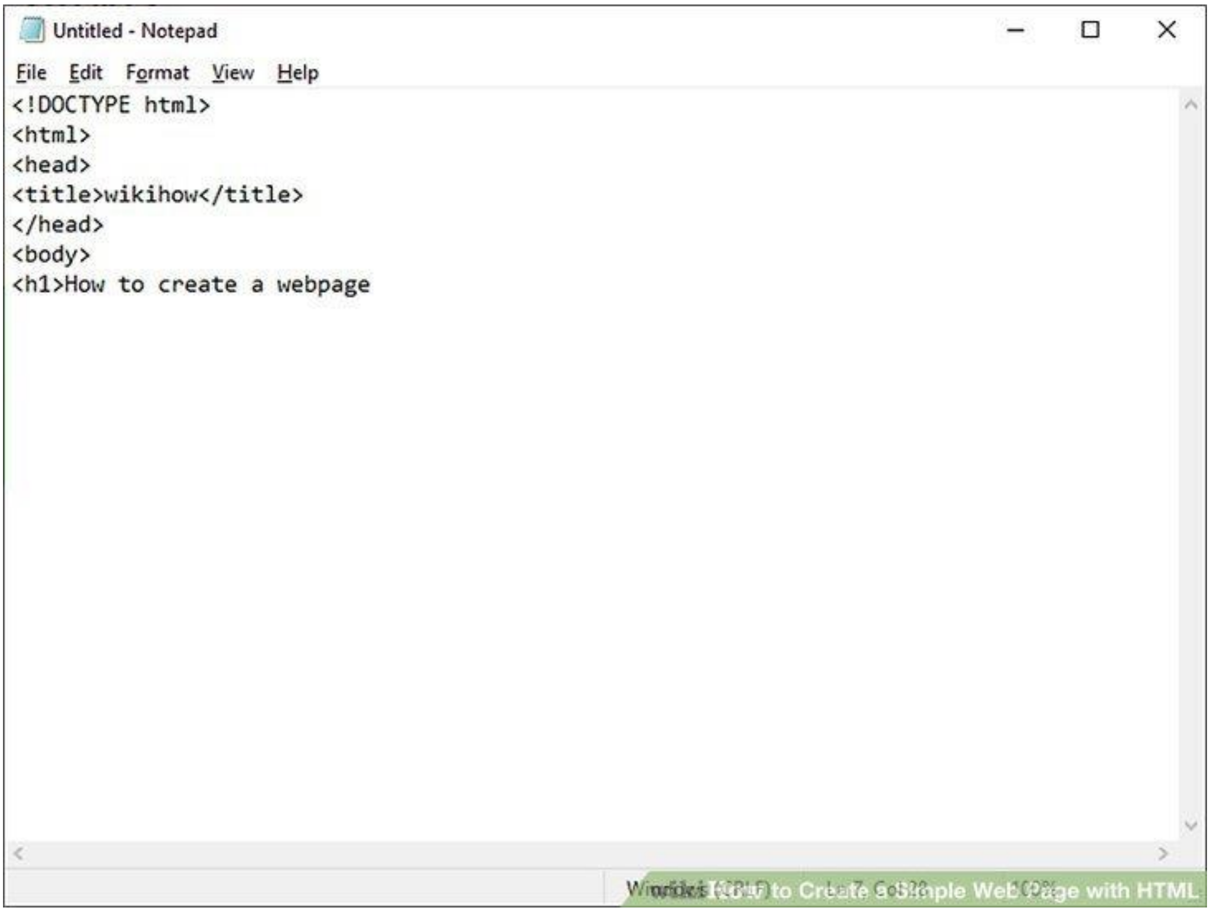


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple Web Page with HTML

```
<!DOCTYPE html>
<html>
<head>
<title>wikihow</title>
</head>
<body>
<h1>
```

3

Type a heading for your page. This can be the title of your page or a greeting.

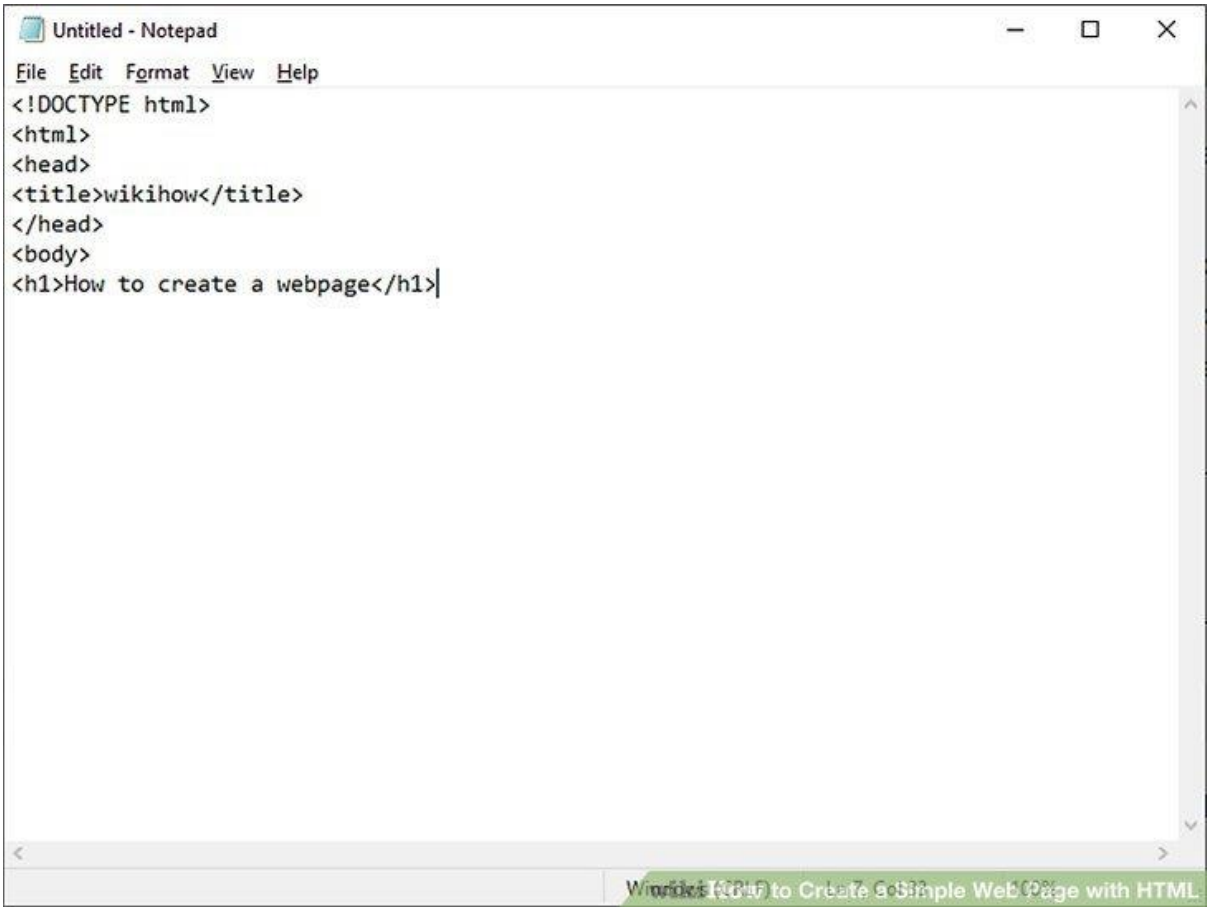

Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple Web Page with HTML

```
<!DOCTYPE html>
<html>
<head>
<title>wikihow</title>
</head>
<body>
<h1>How to create a webpage
```

4

Type `</h1>` after your heading text and press ↵ Enter. This tag closes your heading.

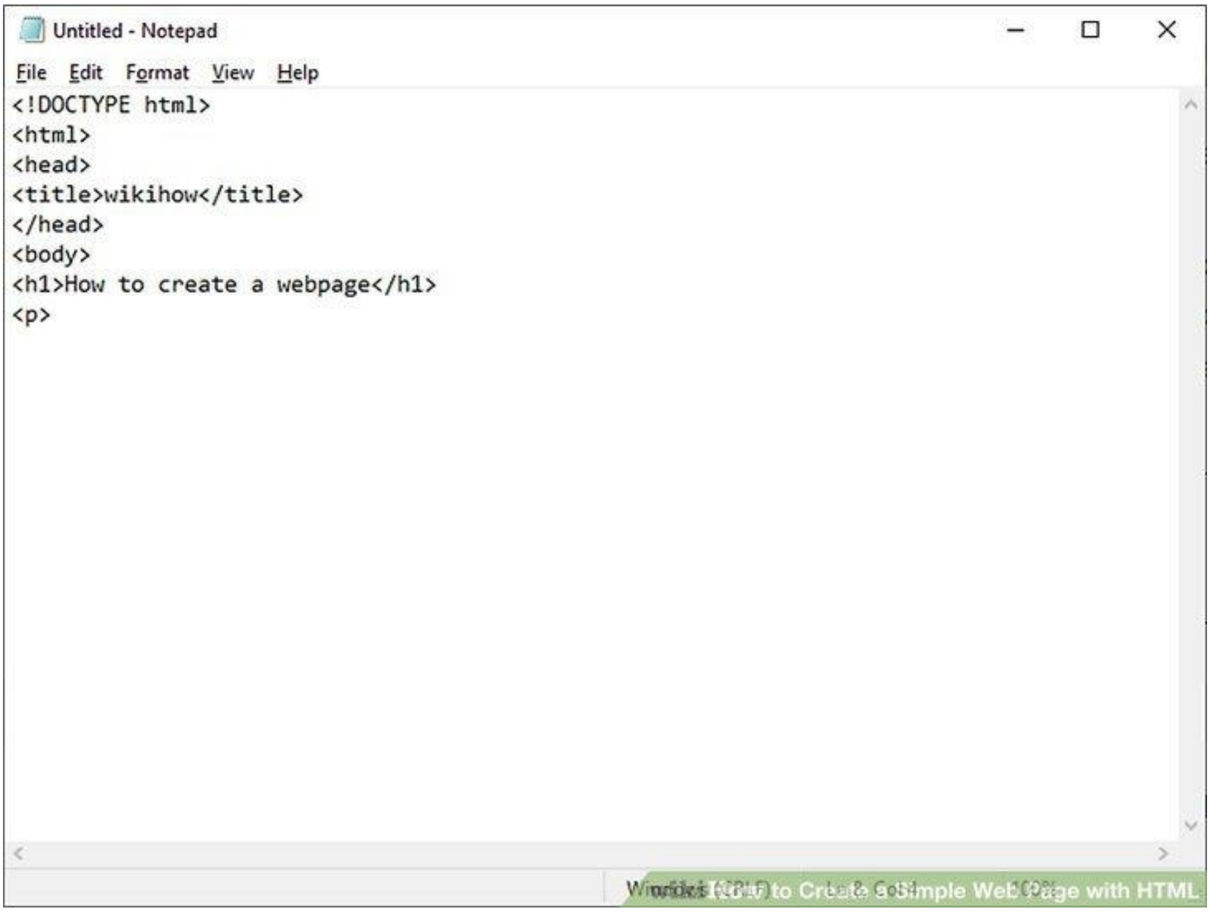
- Add additional headings as you go. There are six different headings that you can create by using the `<h1></h1>` through `<h6></h6>` tags. These create headings of different sizes. For example, to create three different-sized headings in succession, you might write the following:
  - <**h1**>Welcome to My Page!</**h1**>
  - <**h2**>My name is Bob.</**h2**>
  - <**h3**>I hope you like it here.</**h3**>
- The headings shows the priority or importance of the text. But its not necessary to use a higher heading if you want to use any lower heading. One can directly use H3, even if there is no H1 in your post.



```
<!DOCTYPE html>
<html>
<head>
<title>wikihow</title>
</head>
<body>
<h1>How to create a webpage</h1>
```

5

Type `<p>`. This is the tag to open a paragraph. Paragraph text is used to display normal sized text.[6]

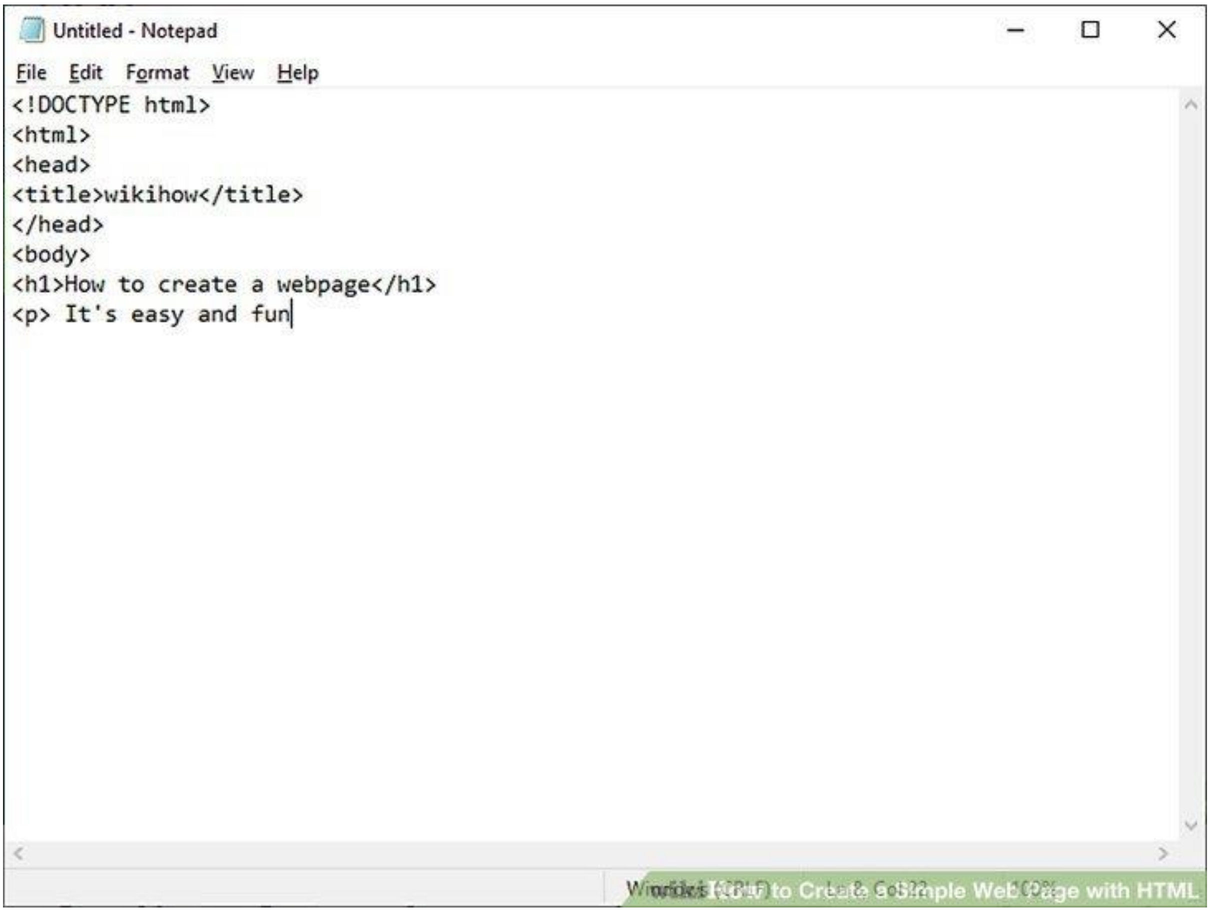


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple Web Page with HTML

```
<!DOCTYPE html>
<html>
<head>
<title>wikihow</title>
</head>
<body>
<h1>How to create a webpage</h1>
<p>
```

6

Type some text. This can be a description for your web page or any other information you wish to share.


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple Web Page with HTML

```
<!DOCTYPE html>
<html>
<head>
<title>wikihow</title>
</head>
<body>
<h1>How to create a webpage</h1>
<p> It's easy and fun
```

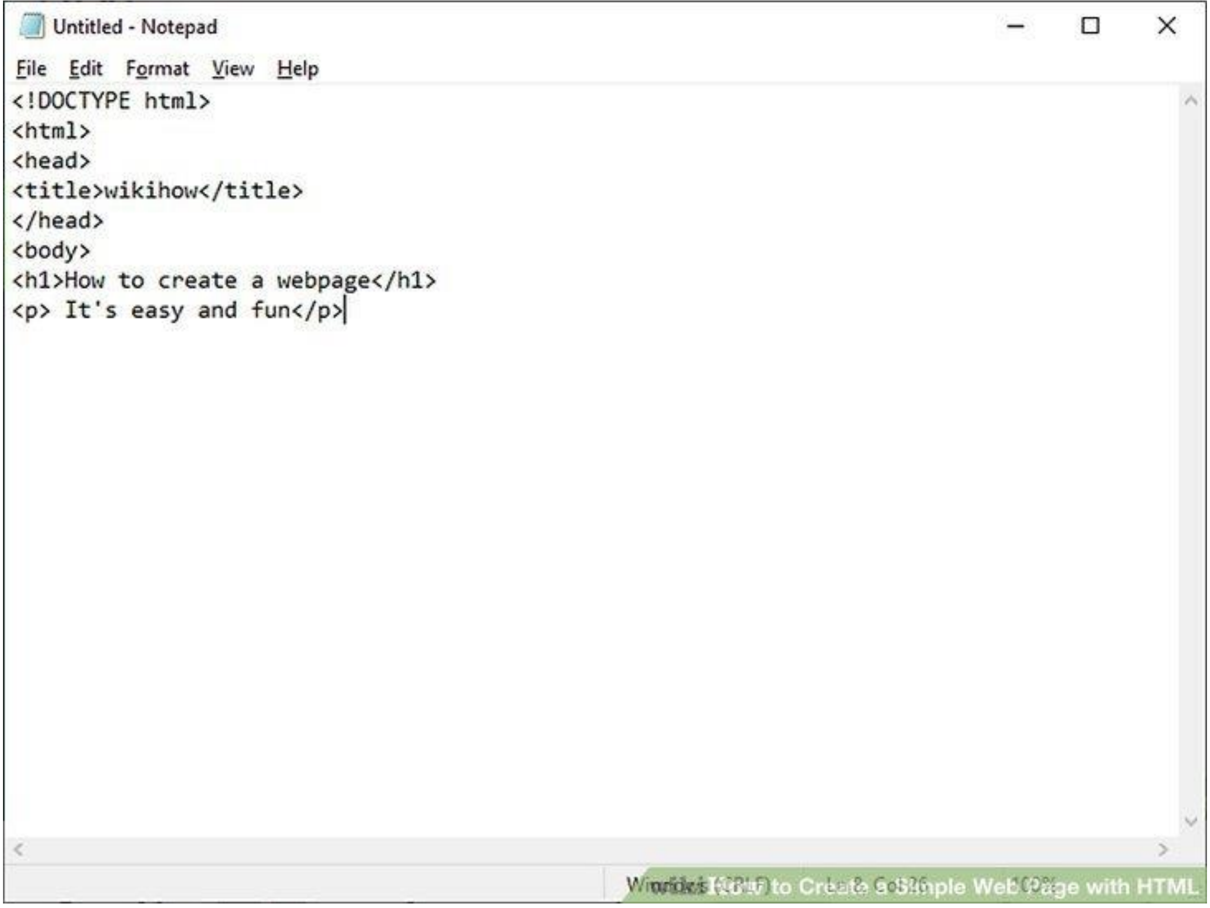
7

Type `</p>` after your text and press ↵ Enter. This the tag to close your paragraph text. The following is an example of paragraph text in HTML:

```
<p>This is my paragraph.</p>
```

- You can add multiple paragraph lines in a row in order to create a series of paragraphs under one heading.
- You can change the color of any text by framing the text with the `<font color="color">` and `</font>` tags. Make sure to type your preferred color into the "color" section (you'll keep the quotes). You can turn any text (e.g., headers) into a different color with this set of tags. For example, to turn a paragraph's text blue, you would write the following code: `<p><font color="blue">Whales are majestic creatures.</font></p>`
- You can add bolds, italics and other text formats using HTML. The following are examples of how you can format text using HTML tags:[7]
  - <b>Bold text</b>
  - <i>Italic text</i>
  - <u>Underlined text</u>
  - <sub>Subscript text</sub>

  - <**sup**>Superscript text</**sup**>
- If you use bold and italic text for emphasis, not just for styling, use the `<strong>` and `<em>` elements instead of `<b>` and `<i>`. This makes your web page easier to understand when using technologies like a screen reader[8] or the reader mode provided in some browsers[9] .



```
<!DOCTYPE html>
<html>
<head>
<title>wikihow</title>
</head>
<body>
<h1>How to create a webpage</h1>
<p> It's easy and fun</p>
```

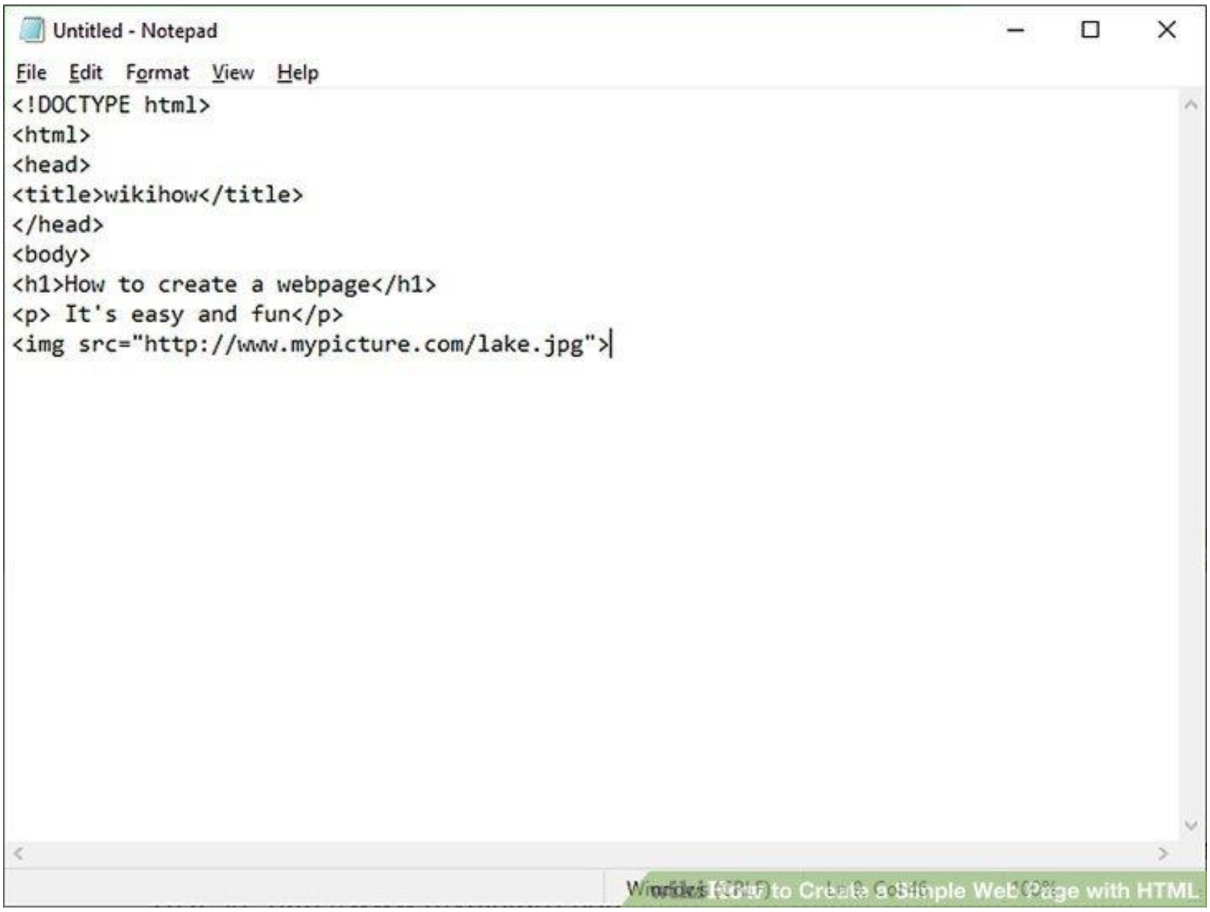
Part 3

Adding Additional Elements to Your HTML

1

Add a picture to your page. You can add an image to your HTML using the following steps:[10]

- Type `<img src=` to open your image tag.
- Copy and paste the image URL after the "=" sign in quotation marks.
- Type `>` after the image url to close your image tag. For example, if the image's URL is "http://www.mypicture.com/lake", you would write the following:
- <**img** src="http://www.mypicture.com/lake.jpg">

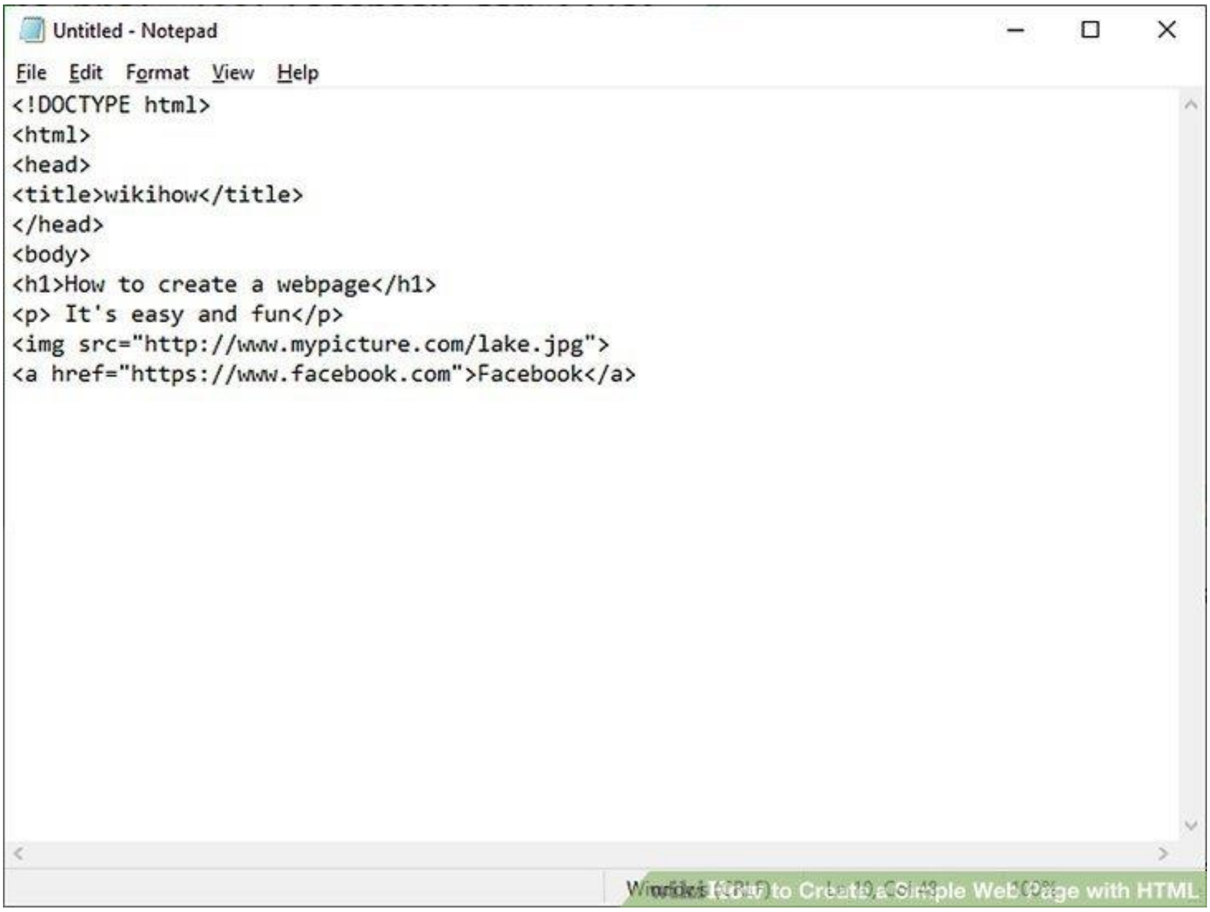

Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple Web Page with HTML

```
<!DOCTYPE html>
<html>
<head>
<title>wikihow</title>
</head>
<body>
<h1>How to create a webpage</h1>
<p> It's easy and fun</p>
<img src="http://www.mypicture.com/lake.jpg">
```

2

Link to another page. You can add a link to your HTML using the following steps:[11]

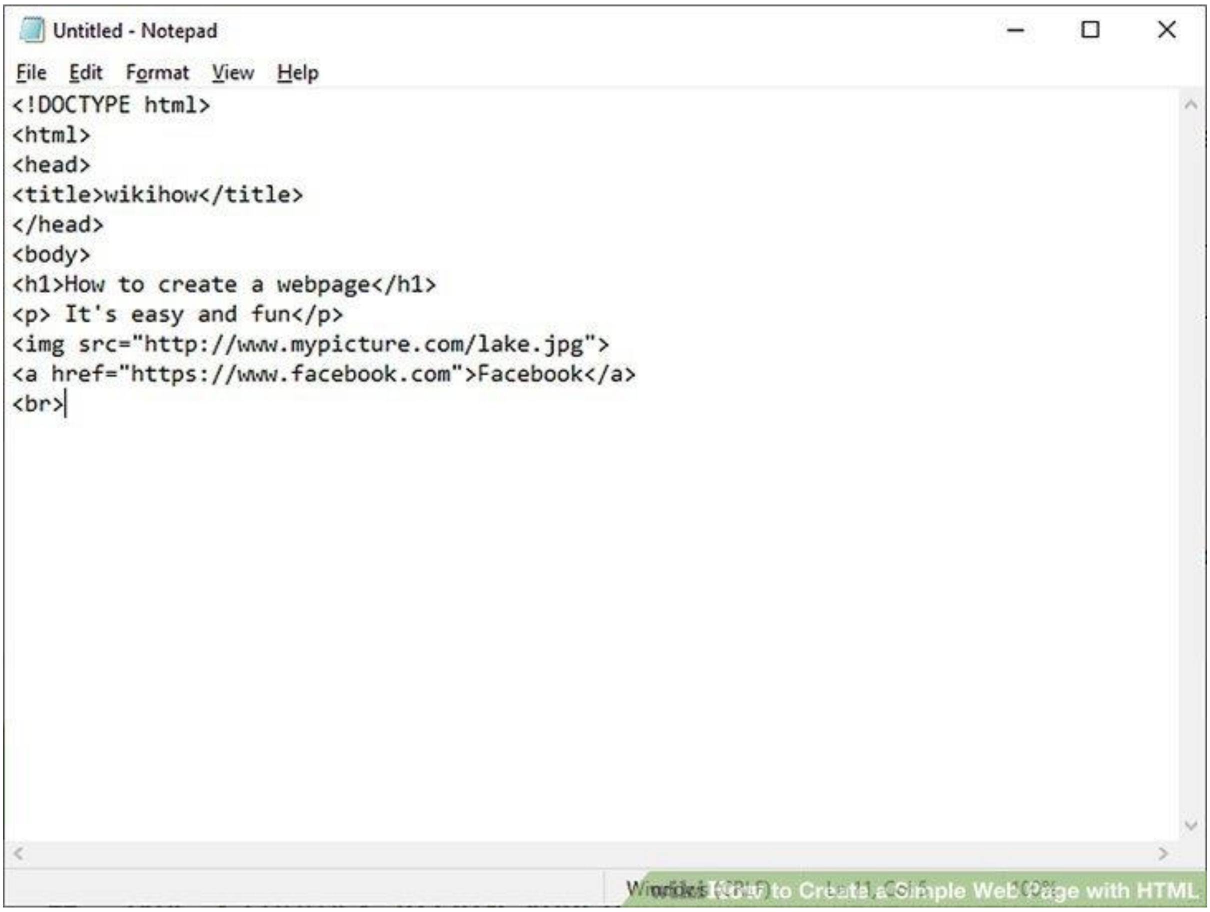
- Type `<a href=` to open your link tag.
- Copy and paste URL after the "=" sign in quotation marks.
- Type `>` after the URL to close the link portion of the HTML.
- Type a name for the link after the closing bracket.
- Type `</a>` after the link name to close the HTML link.[12] The following is an example of a link to Facebook.
- <**a** href="https://www.facebook.com">Facebook</**a**>.


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple Web Page with HTML

```
<!DOCTYPE html>
<html>
<head>
<title>wikihow</title>
</head>
<body>
<h1>How to create a webpage</h1>
<p> It's easy and fun</p>
<img src="http://www.mypicture.com/lake.jpg">
<a href="https://www.facebook.com">Facebook</a>
```

3

Add a line break to your HTML. You can add a line break by typing `<br>` to your HTML. This creates a horizontal line that can be used to divide different sections of your page.[13]


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple Web Page with HTML

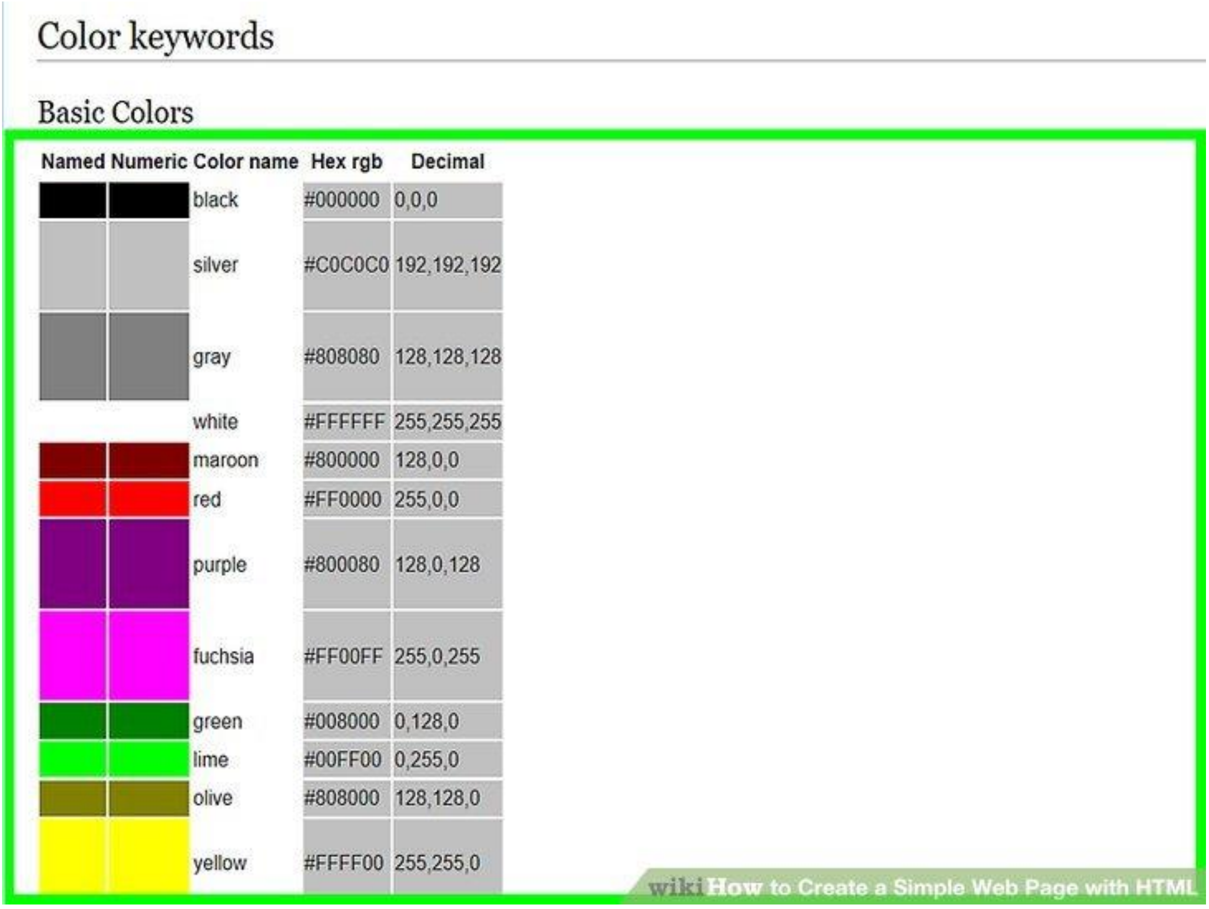
```
<!DOCTYPE html>
<html>
<head>
<title>wikihow</title>
</head>
<body>
<h1>How to create a webpage</h1>
<p> It's easy and fun</p>
<img src="http://www.mypicture.com/lake.jpg">
<a href="https://www.facebook.com">Facebook</a>
<br>
```

Part4

Customizing Colors

1

Check out the list of official HTML color names and codes. The World Wide Web Consortium (W3C) manages an official list of colors that you'll find at https://www.w3.org/wiki/CSS/Properties/color/keywords. Each color has an official name, 6-digit hexadecimal code, and a decimal value. You can use any of these values to add color to elements of your webpage. For this example, we'll use the official color names.



Color keywords

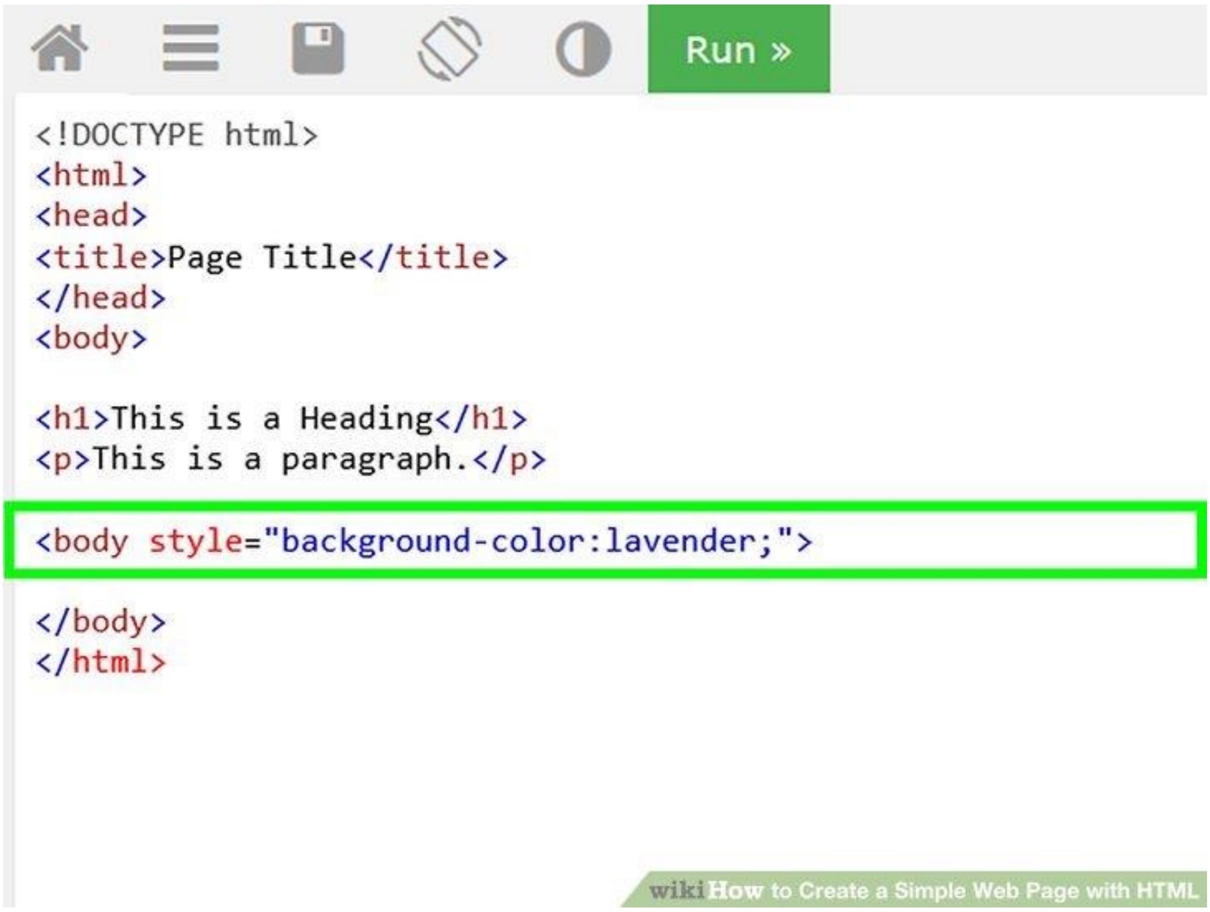
Basic Colors

Named	Numeric	Color name	Hex rgb	Decimal
■	■	black	#000000	0,0,0
■	■	silver	#C0C0C0	192,192,192
■	■	gray	#808080	128,128,128
		white	#FFFFFF	255,255,255
■	■	maroon	#800000	128,0,0
■	■	red	#FF0000	255,0,0
■	■	purple	#800080	128,0,128
■	■	fuchsia	#FF00FF	255,0,255
■	■	green	#008000	0,128,0
■	■	lime	#00FF00	0,255,0
■	■	olive	#808000	128,128,0
■	■	yellow	#FFFF00	255,255,0

2

Set the background color in the `<body>` tag. You'll be doing this by adding the `style` attribute to the tag. Let's say you wanted to make the background color of the entire page `lavender`:

- `<body style="background-color:lavender;">`

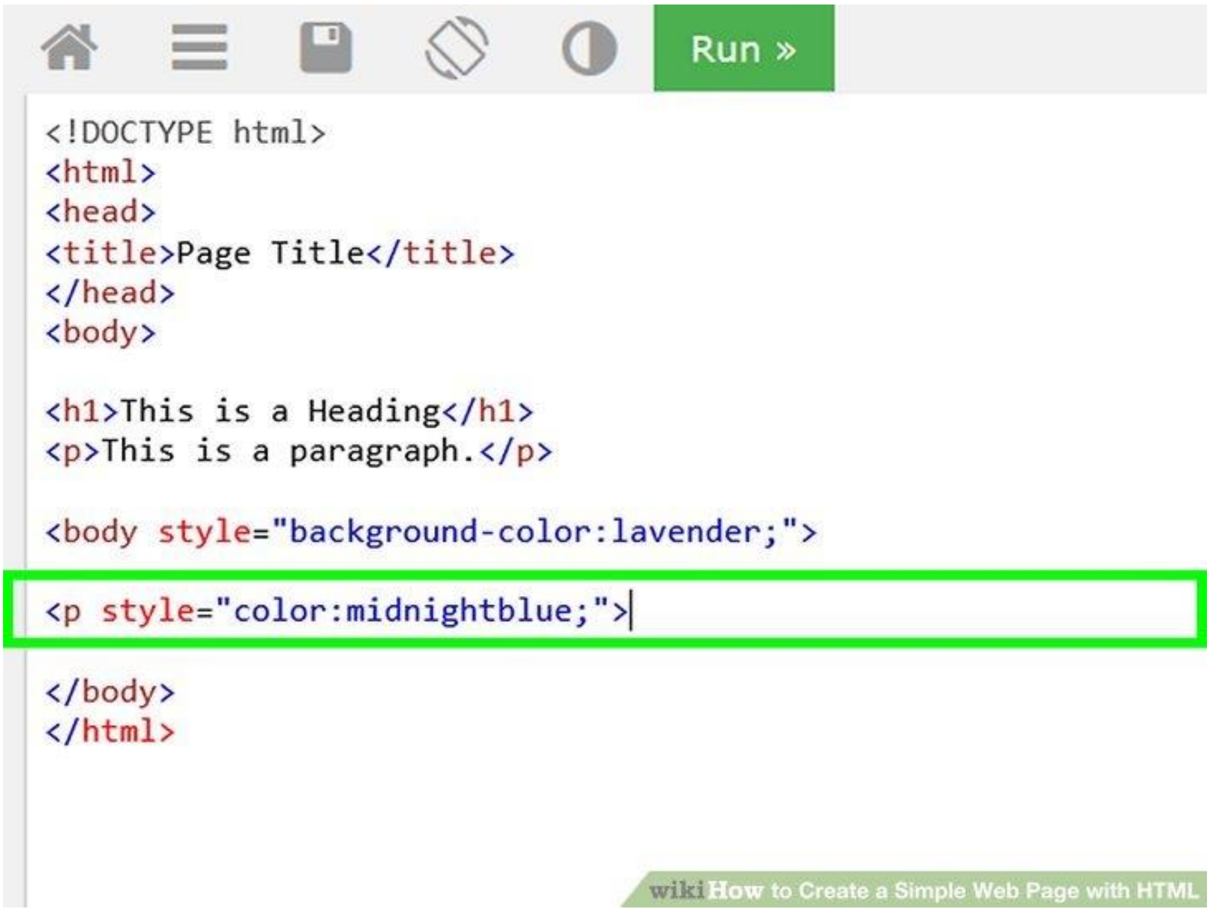


```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

<body style="background-color:lavender;">

</body>
</html>
```

3

Set the text color for any tag. You can also use the `style` attribute to specify which color you'd like all text within a particular tag to be. For example, let's say you wanted to make the text in one of your `<p>` tags `midnightblue`:[14]

- `<p style="color:midnightblue;">`
- The color change will only affect the text within that `<p>` tag. If you start another `<p>` tag later that should also be `midnightblue`, you'll need to set the style attribute there as well.

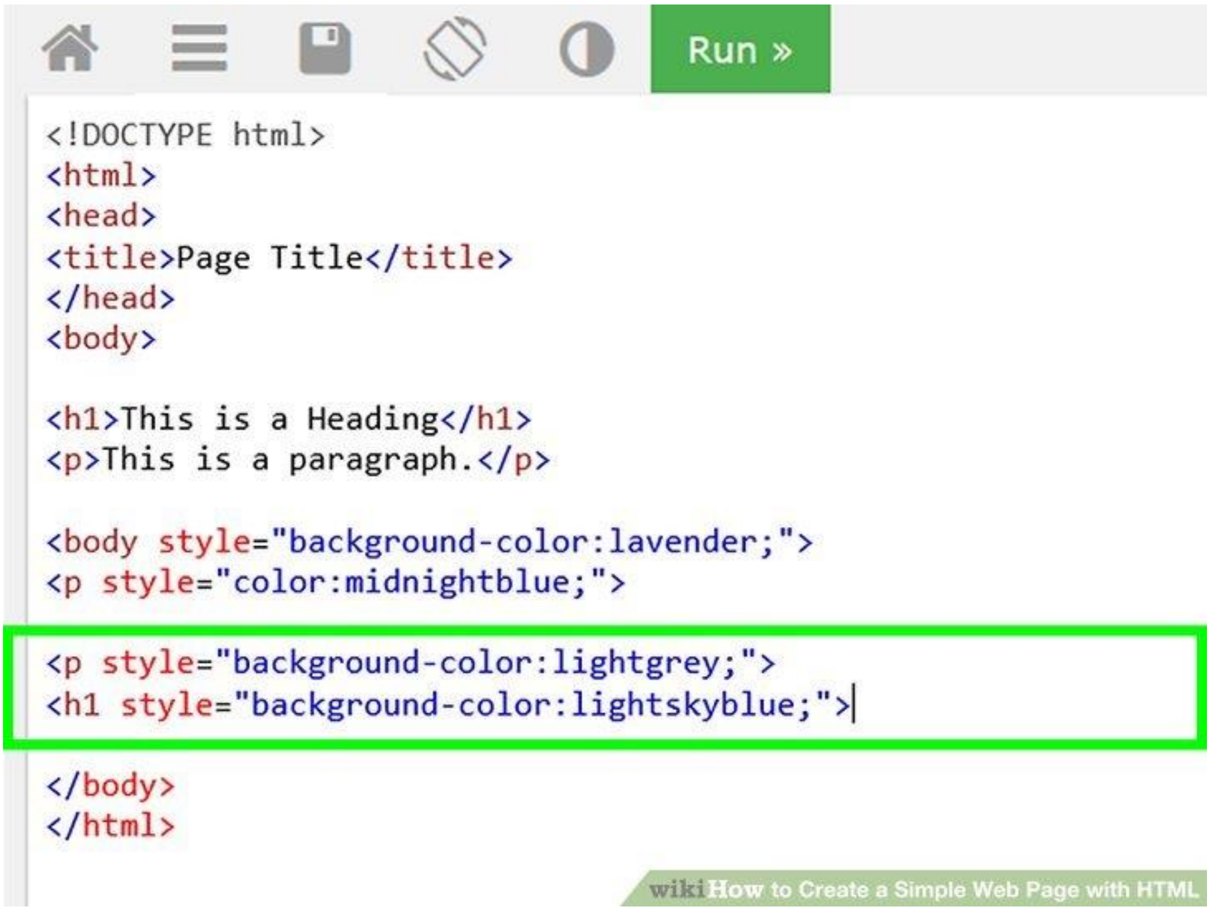


```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

<body style="background-color:lavender;">

<p style="color:midnightblue;">

</body>
</html>
```

4

Set the background color for a header or paragraph. Similar to how you set the background color for the body tag, you can also set background colors for other tags. Let's say you wanted to make the background color of an `<p>` `lightgrey`, and the background color of an H1-style header `lightskyblue`, you'd use:

- `<p style="background-color:lightgrey;">`
- `<h1 style="background-color:lightskyblue;">`



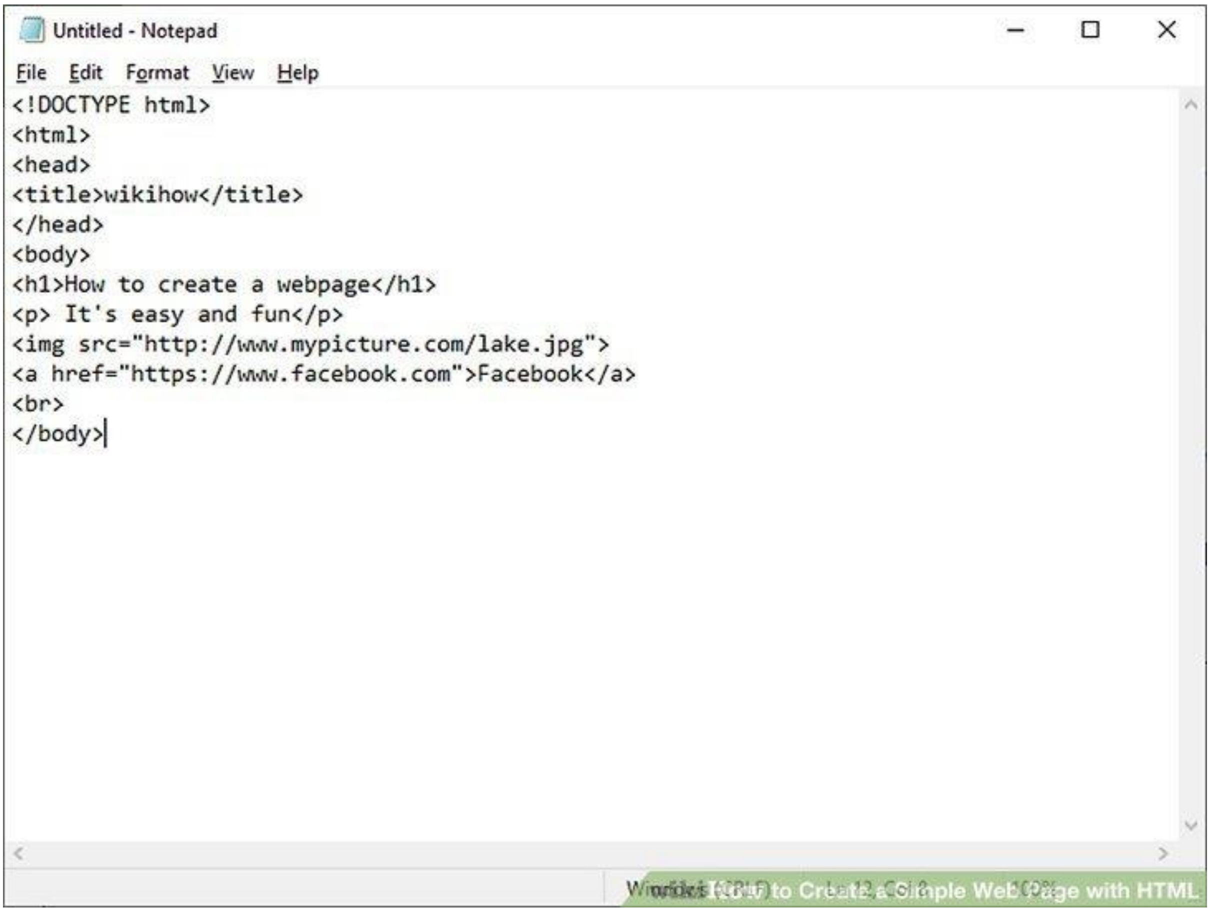
```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

<body style="background-color:lavender;">
<p style="color:midnightblue;">

<p style="background-color:lightgrey;">
<h1 style="background-color:lightskyblue;">

</body>
</html>
```

Part5

Closing Your HTML Document

1

Type `</body>` to close your body. After you have finished adding all your text, images and other elements to the body of your HTML document, add this tag at the bottom of your HTML document to close the body of your HTML document.


Untitled - Notepad
File Edit Format View Help
wikiHow How to Create a Simple Web Page with HTML

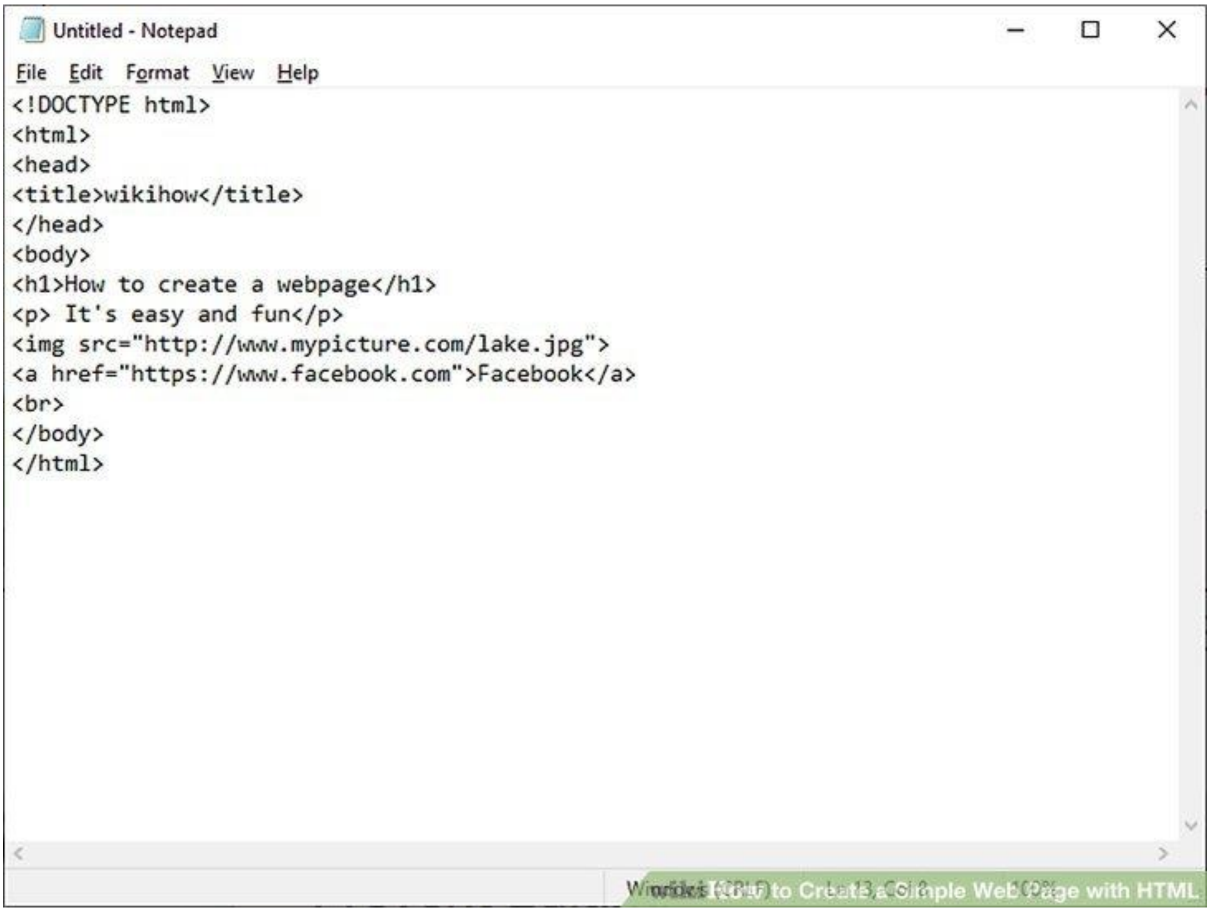
```
<!DOCTYPE html>
<html>
<head>
<title>wikihow</title>
</head>
<body>
<h1>How to create a webpage</h1>
<p> It's easy and fun</p>
<img src="http://www.mypicture.com/lake.jpg">
<a href="https://www.facebook.com">Facebook</a>
<br>
</body>
```

2

Type `</html>` to close your HTML document. This tag goes below the tag to close your HTML body at the end of your HTML document. This tells the web browser there is no more HTML code after this tag. Your entire HTML document should look something like this:

```
<!DOCTYPE html>
<html>

<head>
<title>wikiHow Fan Page</title>
</head>

<body>

<h1>Welcome to My Page!</h1>
<p>This is a fan page for wikiHow. Make yourself at home!</p>

<h2>Important Dates</h2>
<p><i>January 15, 2019</i> - wikiHow's Birthday</p>
```

```
<h2>Links</h2>
<p>Here is a link to wikiHow: <a href="http://www.wikihow.com">wikiHow</a></p>

</body>
</html>
```



Untitled - Notepad
File Edit Format View Help
How to Create a Simple Web Page with HTML

```
<!DOCTYPE html>
<html>
<head>
<title>wikihow</title>
</head>
<body>
<h1>How to create a webpage</h1>
<p> It's easy and fun</p>
<img src="http://www.mypicture.com/lake.jpg">
<a href="https://www.facebook.com">Facebook</a>
<br>
</body>
</html>
```

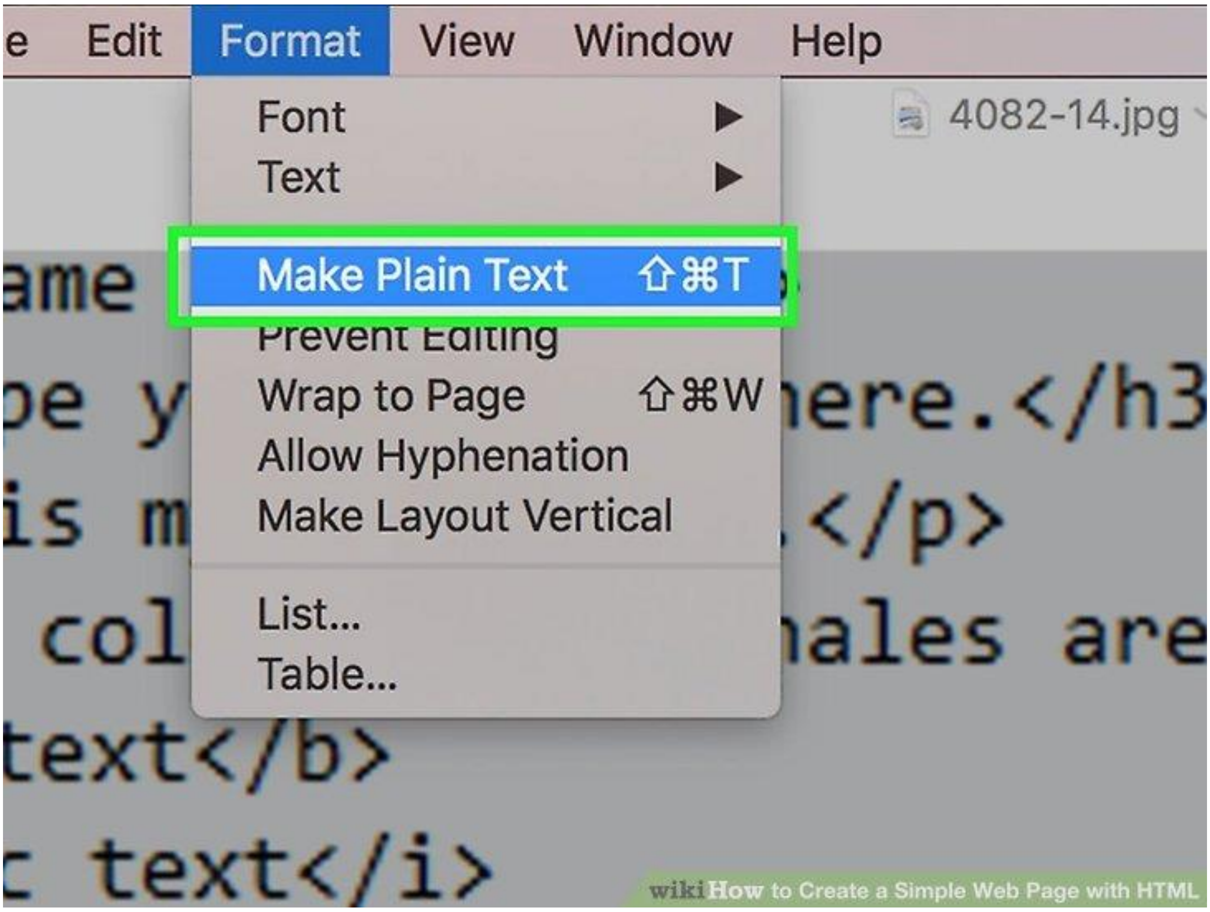
Part6

Saving and Opening Your Web Page

1

Convert your document to plain text (Mac users only). Click the **Format** menu item at the top of the screen, then click **Make Plain Text** in the resulting drop-down menu.[15]

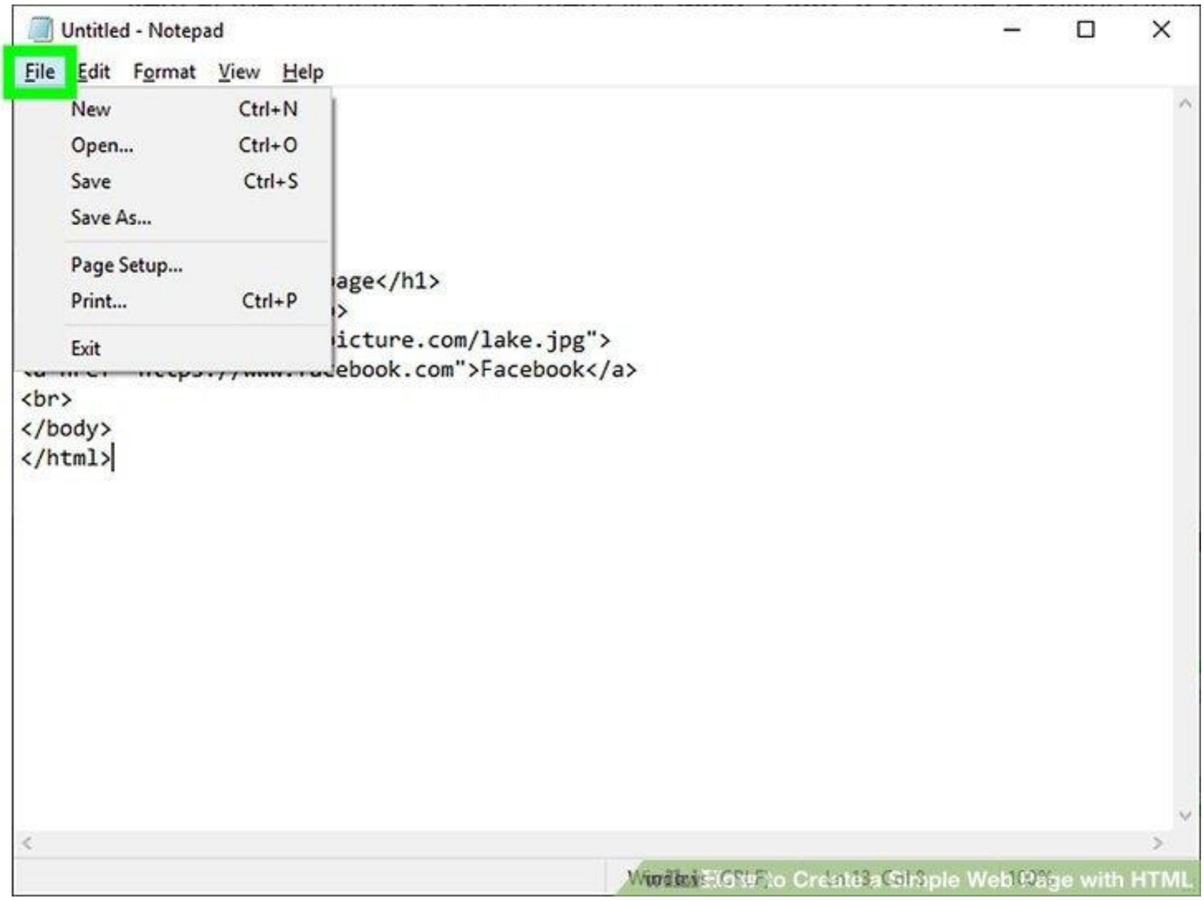
- This step is neither necessary nor possible on Windows.



```
ame
pe y          here.</h3
is m          .</p>
 col          nales are
text</b>
c text</i>
```

2

Click File. It's in the menu bar at the top of the screen.

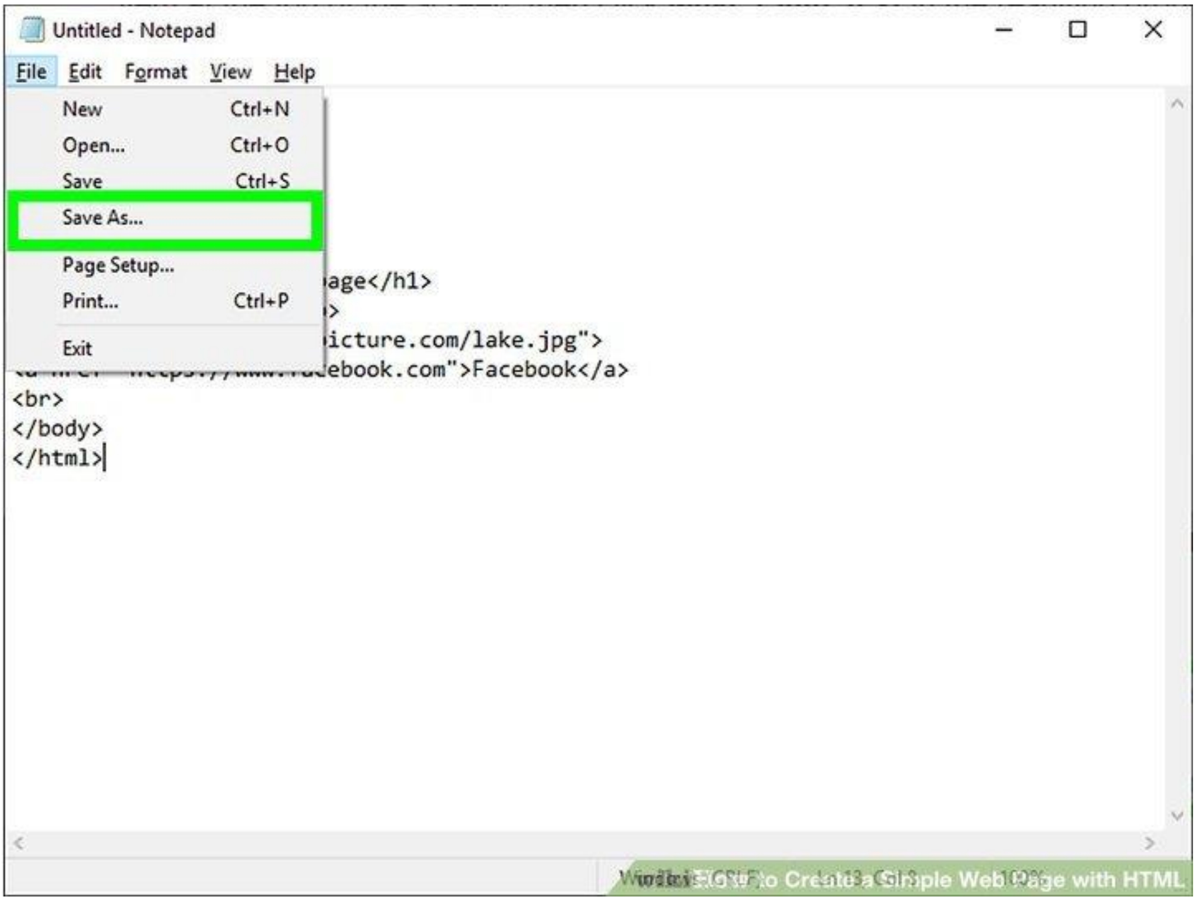


```
                    age</h1>
                    >
                    icture.com/lake.jpg">
                    ebook.com">Facebook</a>
<br>
</body>
</html>
```

3

Click Save as. It's in the drop-down menu below "File".

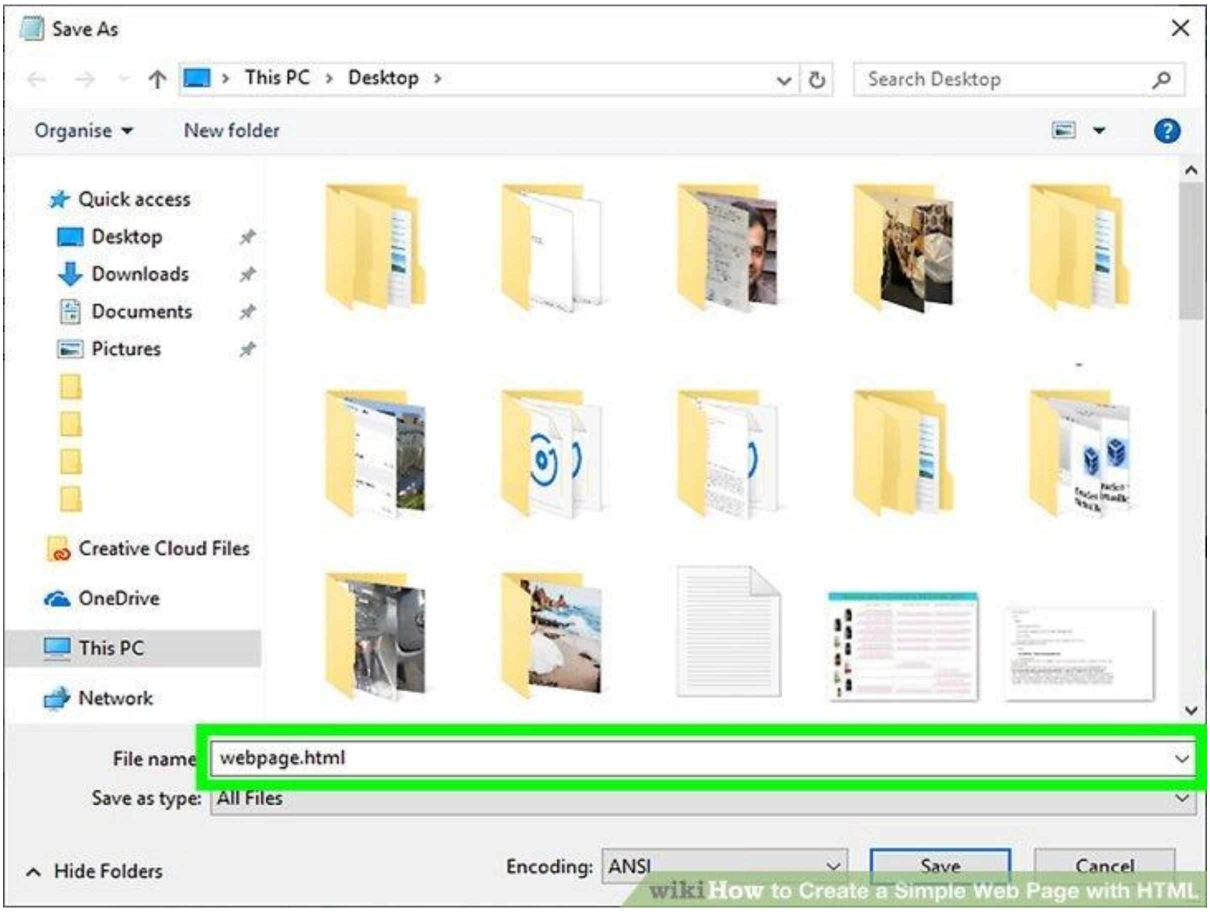
- Alternatively, you can press Ctrl + S (Windows) or ⌘ Command + S (Mac) to do so.



```
age</h1>
>
icture.com/lake.jpg">
ebook.com">Facebook</a>
<br>
</body>
</html>
```

4

Enter a name for your HTML document. Type whatever you want to name your document into the "File name" (Windows) or "Name" (Mac) text box.



File name: webpage.html

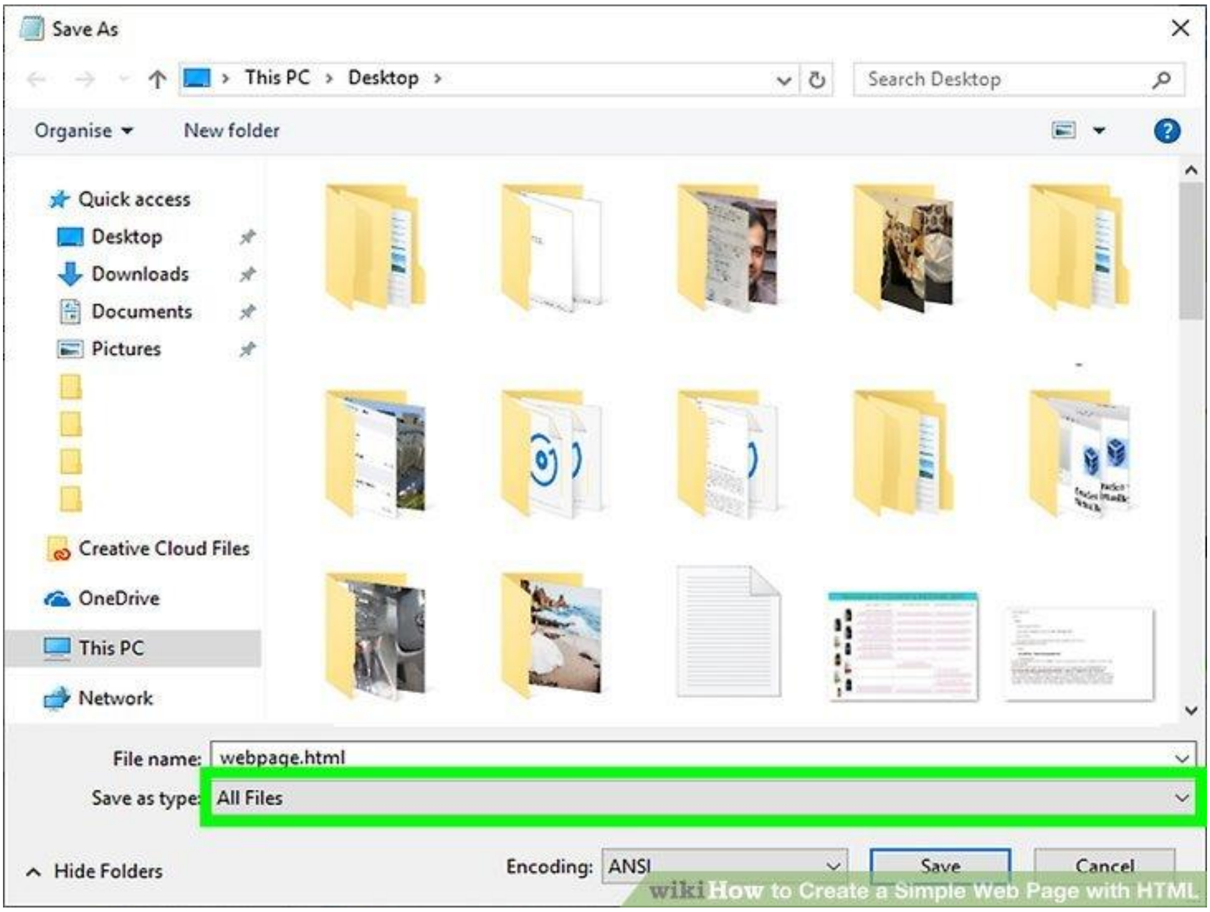
Save as type: All Files

Encoding: ANSI

5

Change the document's file type. You'll need to change the document from a text file to an HTML file. Use the following steps to change the file type:

- *Windows* - Click the "Save as type" drop-down box, click **All Files**, and then type `.html` at the end of the file's name.
- *MacOS* - Replace the `.txt` at the end of the file's name with `.html` instead.
- *ChromeOS* - Click the "Save as" button. Name the file with `.html` at the end. The beginning is up to you.



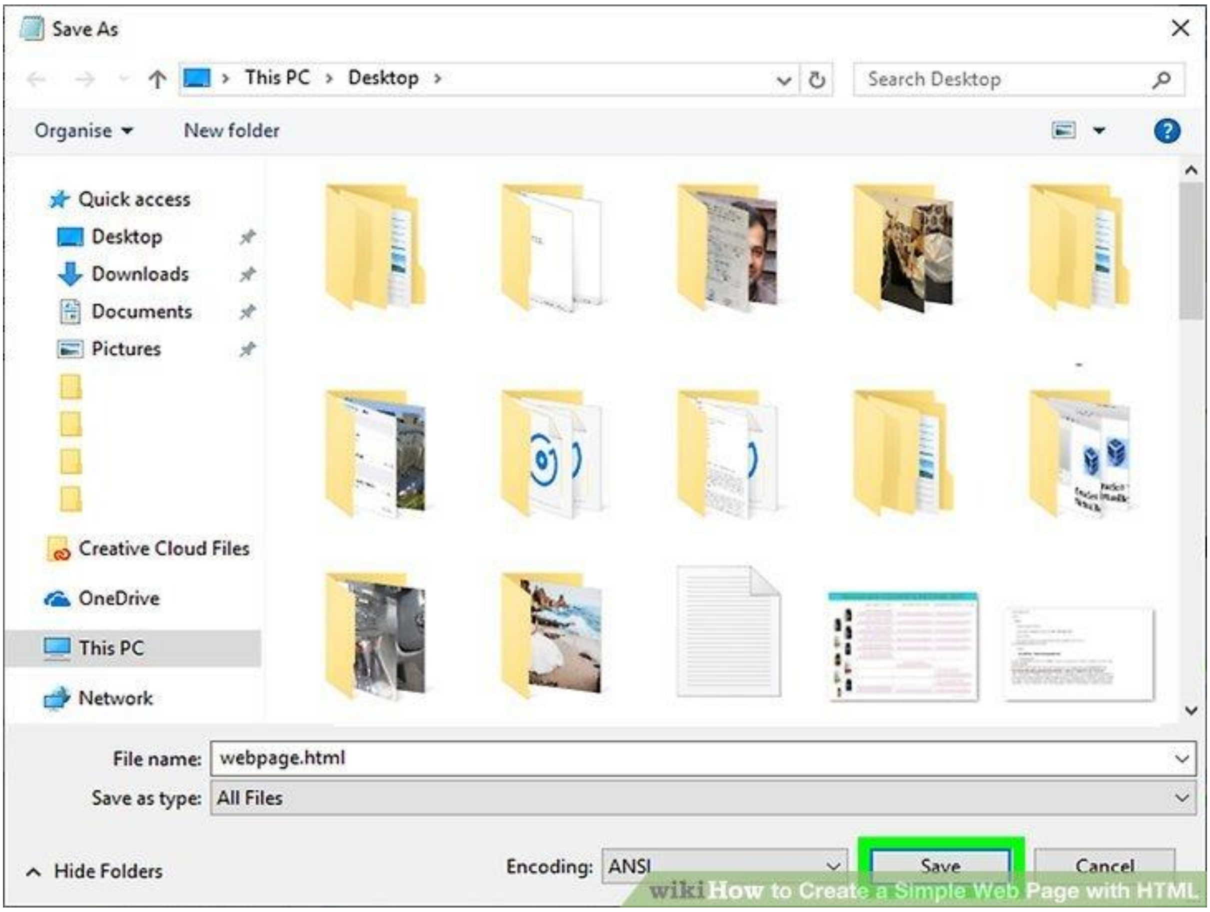
File name: webpage.html

Save as type: All Files

6

Click Save It's at the bottom of the window. Doing so will create an HTML file.

- HTML files typically open with your default web browser.



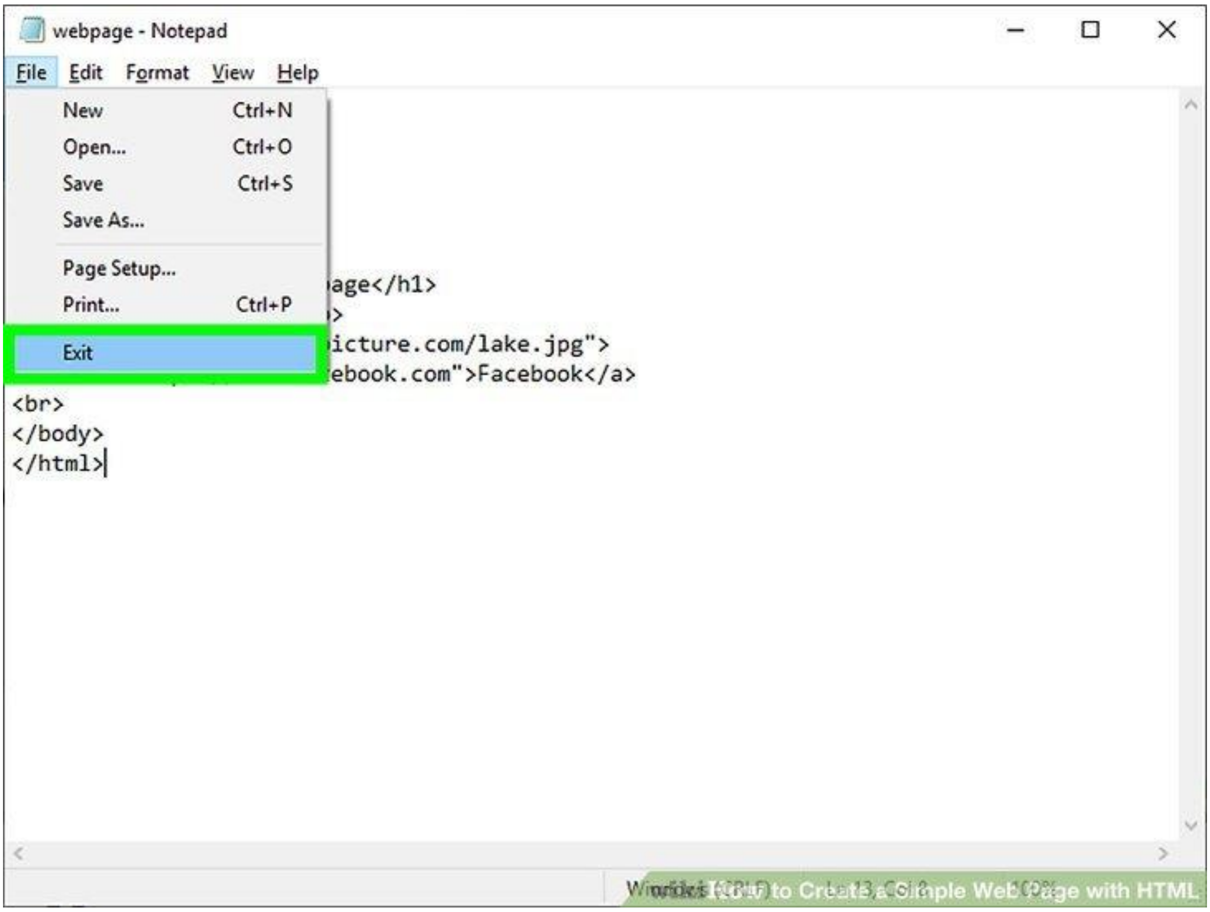
File name: webpage.html

Save as type: All Files

Encoding: ANSI

7

Close your text editor. At this point, you're ready to open your HTML file in your browser so that you can view your web page.

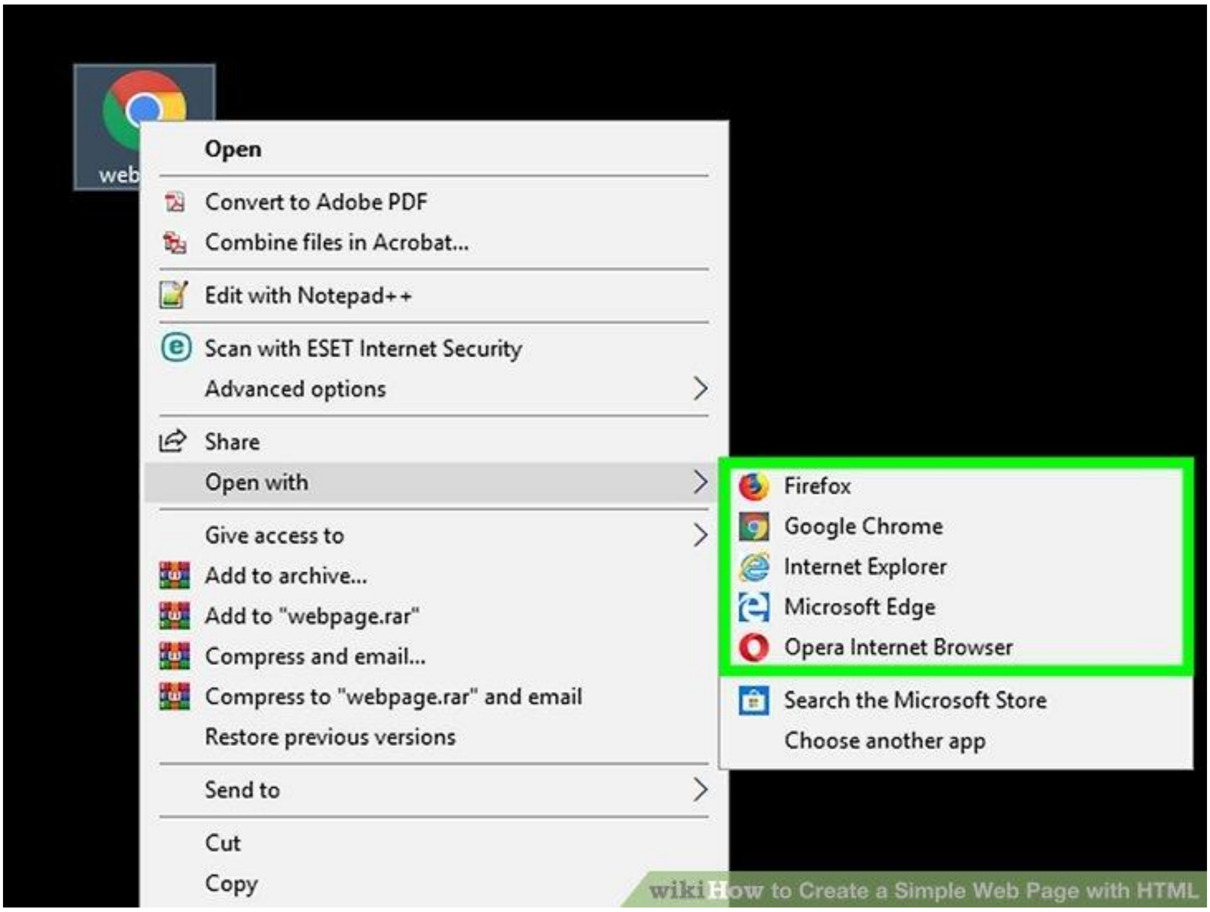


```
age</h1>
>
icture.com/lake.jpg">
ebook.com">Facebook</a>
<br>
</body>
</html>
```

8

Open the HTML document with your browser. In most cases, you'll be able to double-click the HTML document to do this. If double-clicking the document results in an error, do the following:

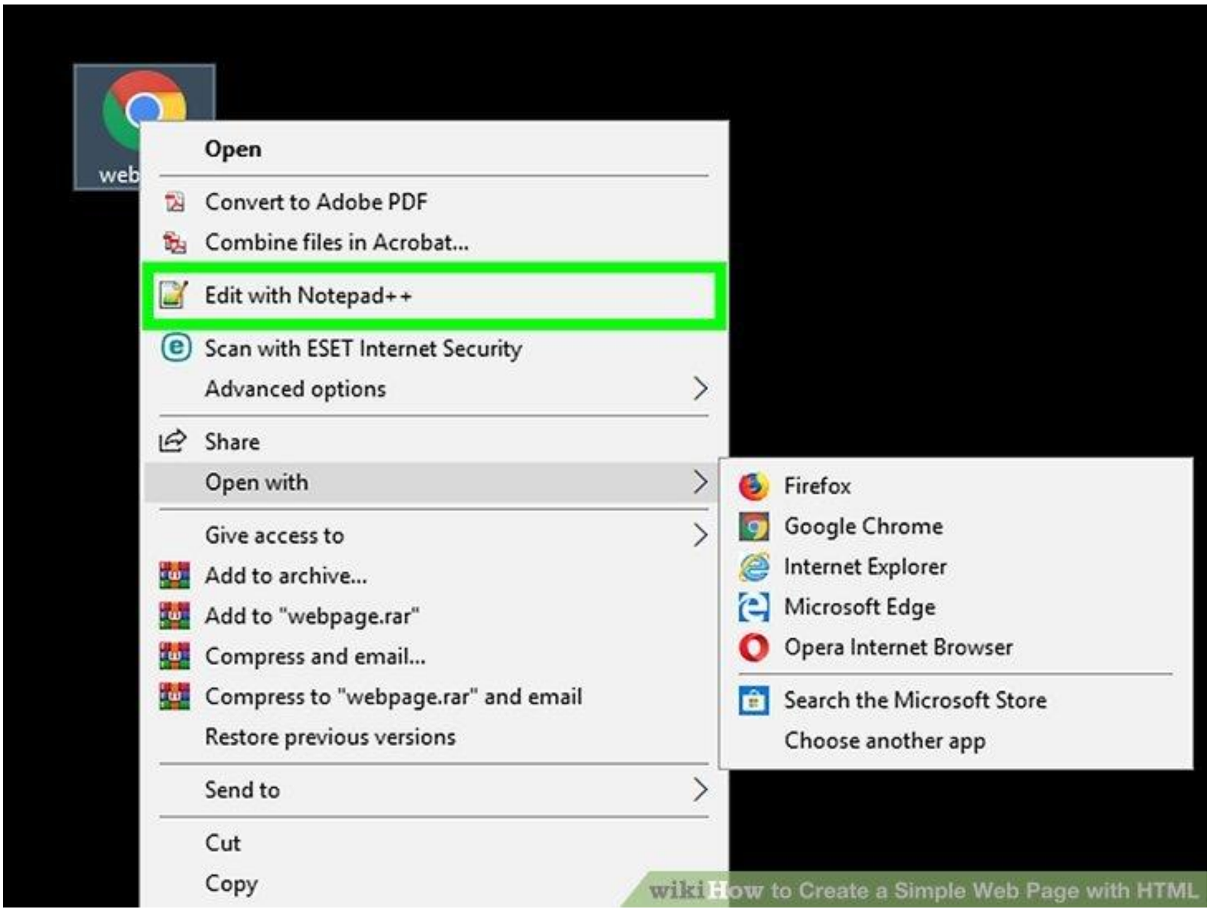
- *Windows* - Right-click the document, select **Open with**, and click your preferred browser.
- *Mac* - Click the document once, click **File**, select **Open With**, and click your preferred browser.



9

Edit the HTML document if needed. You may notice an error in your HTML page. To change it, you can edit the HTML document's text:

- On Windows, you can right-click the document and click **Edit** in the resulting drop-down menu (if you have Notepad++ installed, this will say **Edit with Notepad++** instead).
- On Mac, you'll want to click the document to select it, click **File**, select **Open With**, and click **TextEdit**. You can also drag the document into TextEdit.
- On Chromebook, close the Text app, open Files, find your file, and then click on it.



Sample HTML

HTML Cheat Sheet

Bolded text

```
<b>Text goes here</b>
<strong>Text goes here</strong>
```

Italicized text

```
<i>Text goes here</i>
<em>Text goes here</em>
```

Underlined text

```
<u>Text goes here</u>
```

Changing font color

```
<font color=”red”>Text goes here</font>
<font color=”#0099ff”>Text goes here</font>
```

Changing font size

8 <font size=”12px”>Text goes here</font>
9 <font size=”large”>Text goes here</font>

Adding links

10 <a href=”http://www.URLgoeshere.com” target=”_blank”>Link goes here</a>
   a. target=_blank indicates that the link will open in a new browser window; you may remove it if you’d like the link to open in the same window

Adding images

11 <img src=”http://imageURLgoeshere.com/image.jpg” border=”1px #000 solid” align=”right” />
   a. border=”1px #000 solid” will put a black border around your image generated by each visitor’s web browser. The first value (1px) refers to the thickness; the second value (#000) refers to the color, which can be replaced with any web-based color name or RGB hex code; the third value (solid) refers to the pattern of the line, which can be replaced with dotted or dashed
   b. align=”right” will manipulate the image to align with the text in a specified way; you may replace right with left, center, or justify

Adding a horizontal divider between sections on your webpage

12 <hr />

Adding a line break

13 <br />

Making a nested, bulleted list

14.<ul>
15.<li>Section A</li>
   1. <ul>
   2. <li>Section A Part 1</li>
   3. <li>Section A Part 2</li>
      1. <ul>
      2. <li>Section A Part 2.1</li>
      3. <li>Section A Part 2.2</li>

        4. <li>Section A Part 2.3</li>
        5. </ul>
    4. <li>Section A Part 3</li>
    5. <li>Section A Part 4</li>
    6. </ul>
16. <li>Section B</li>
17. <li>Section C</li>
    1. <ul>
    2. <li>Section C Part 1</li>
    3. <li>Section C Part 2</li>
    4. </ul>
18. <li>Section D</li>
19. </ul>

Making a numbered list

20. Replace <ul> with <ol> and </ul> with </ol> in the code above
21. You can also combine numbers and bullets within the same nested list, depending on whether you choose to use <ul> </ul> or <ol> </ol> for each list level

Making paragraphs

22. <p>A whole paragraph of text goes here!</p>
23. You may modify the formatting of each paragraph by inserting CSS into your HTML code. For example:
    1. <p style=”font-size:14px;color:green;font-weight:bold;align=center;”>This whole paragraph of text will be centered, green and in 14px.</p>
24. You can add as many or as few of the following CSS modules as you’d like to the <p style=””> </p> tag demonstrated above
    1. font-size:##px; modifies the size of the text; px can be replaced with pt, em, % or any length unit (cm, in, etc). In lieu of numbers, you can also write font-size:large; and replace large with xx-small, x-small, small, medium (default for most browsers), large, x-large, xx-large, larger, smaller.
    2. font-family:Arial; modifies the font type of the text; can be replaced with the name of any font, but if the user’s computer does not have the font downloaded, then it will revert to the browser’s default. If the font name has spaces in it, e.g., Times New Roman, it should be placed within quotation marks.
    3. color:######; or color:red; changes the color of the text. For a list of acceptable web color names, click here.

4. font-weight:bold; determines whether the text is bolded or normal; bold can be replaced with normal.
5. font-style: italic; determines whether the text is italicized or not; italic can be replaced with normal.
6. text-decoration:underline; adds an underline to the text; underline can also be replaced with overline for a line over the text, line-through for a line through the text, and none.
7. text-transform:uppercase; transforms the casing of the text; uppercase (all caps) can be replaced with capitalize (first letter of each word is capitalized), lowercase (all lowercase letters), or none.
8. text-indent:5em; will indent the first line of the paragraph by the specified amount; any unit acceptable for font-size can be used for text-indent.

Sample Webpage with HTML

```
<p style="color:#003399;font-size:18px;font-weight:bold;text-align:center;">Welcome to My Webpage!</p>

<p style=”color:#000;font-size:14px;font-weight:normal;text-align:left;”>You have stumbled upon the lovely webpage of Wanda WikiHow. This is my little nook on the Internet, featuring my artwork, writing, graphic designs, and future project ideas. Please feel free to take a look around using the navigation menu to the left, and to leave a comment with your thoughts! If you’d like to send me an email, head to the <a href=”/contactme.html” target=”_blank”>contact</a> page. I’d be more than happy to return your correspondence!</p>

<hr />

<p style=”color:#454545;font-size:16px;font-weight:normal;font-style:italic;text-align:left;”>Latest Updates</p>

<p style=”color:#000;font-size:14px;font-weight:normal;text-align:left;”>

<strong>March 25, 2012:</strong> I uploaded three <a href=”/artwork/watercolor.html” target=”_blank”>watercolor</a> pieces to my portfolio.</p>

<p style=”color:#000;font-size:14px;font-weight:normal;text-align:left;”>

<strong>March 14, 2012:</strong> I had a random moment of inspiration in the middle of the night, which I took advantage of to produce my newest poem, <em>In the Night Sky</em>. Click <a href=”/writing/poems.html” target=”_blank”>here</a> to enjoy!</p>
```

<p style=”color:#000;font-size:14px;font-weight:normal;text-align:left;”>

<strong>February 14, 2012:</strong> In honor of Valentine’s Day, I have uploaded two <a href=”/writing/poems.html” target=”_blank”>poems</a> and three <a href=”/artwork/sketches.html” target=”_blank”>sketches</a> with a subtle but nuanced romantic bent. I challenged myself to think of love outside of the traditional cliche conceptions so those without significant others can still appreciate this lovely holiday. We all have so much more love in our lives than the stereotypical portrayal of romantic affinity; we should open our eyes to all of the different ways in which we are blessed and appreciate it!</p>

Community Q&A

- **Question**

How do I upload my webpage and make it public?

Community Answer

You can set up your own server, but I recommend buying web hosting from some of the available hosting companies. There are also free hosts out there, but they would put their ads on your webpage.

- **Question**

Can I create a web page using Notepad?

Community Answer

Yes. Write the code and then press edit-save and then call it what ever you want. After you called it something, you have to type .html at he end. Save and use as needed.

- **Question**

Can I create an interactive design of a website using only HTML?

Community Answer

Yes, you can put some pictures on there and a background as well.

Color keywords

Basic Colors

Named	Numeric	Color name	Hex rgb	Decimal
⬛	⬛	black	#000000	0,0,0
▨	▨	silver	#C0C0C0	192,192,192

■	■			
■	■	gray	#808080	128,128,128
		white	#FFFFFF	255,255,255
■	■	maroon	#800000	128,0,0
■	■	red	#FF0000	255,0,0
■	■	purple	#800080	128,0,128
■	■	fuchsia	#FF00FF	255,0,255
■	■	green	#008000	0,128,0
■	■	lime	#00FF00	0,255,0
■	■	olive	#808000	128,128,0
■	■	yellow	#FFFF00	255,255,0
■	■	navy	#000080	0,0,128
■	■	blue	#0000FF	0,0,255
■	■	teal	#008080	0,128,128
■	■	aqua	#00FFFF	0,255,255

Extended colors

Named	Numeric	Color name	Hex rgb	Decimal
■	■	aliceblue	#f0f8ff	240,248,255
■	■	antiquewhite	#faebd7	250,235,215
■	■	aqua	#00ffff	0,255,255
■	■	aquamarine	#7fffd4	127,255,212
■	■	azure	#f0ffff	240,255,255
■	■	beige	#f5f5dc	245,245,220

⬜	⬜	bisque	#ffe4c4	255,228,196
⬛	⬛	black	#000000	0,0,0
⬜	⬜	blanchedalmond	#ffebcd	255,235,205
🟦	🟦	blue	#0000ff	0,0,255
🟪	🟪	blueviolet	#8a2be2	138,43,226
🟫	🟫	brown	#a52a2a	165,42,42
🟫	🟫	burlywood	#deb887	222,184,135
🟦	🟦	cadetblue	#5f9ea0	95,158,160
🟩	🟩	chartreuse	#7fff00	127,255,0
🟧	🟧	chocolate	#d2691e	210,105,30
🟧	🟧	coral	#ff7f50	255,127,80
🟦	🟦	cornflowerblue	#6495ed	100,149,237
⬜	⬜	cornsilk	#fff8dc	255,248,220
🟥	🟥	crimson	#dc143c	220,20,60
🟦	🟦	cyan	#00ffff	0,255,255
🟦	🟦	darkblue	#00008b	0,0,139
🟦	🟦	darkcyan	#008b8b	0,139,139
🟨	🟨	darkgoldenrod	#b8860b	184,134,11
▨	▨	darkgray	#a9a9a9	169,169,169
🟩	🟩	darkgreen	#006400	0,100,0
	▨	darkgrey	#a9a9a9	169,169,169
🟨	🟨	darkkhaki	#bdb76b	189,183,107
🟪	🟪	darkmagenta	#8b008b	139,0,139

🟩	🟩	darkolivegreen	#556b2f	85,107,47
🟧	🟧	darkorange	#ff8c00	255,140,0
🟪	🟪	darkorchid	#9932cc	153,50,204
🟥	🟥	darkred	#8b0000	139,0,0
🟧	🟧	darksalmon	#e9967a	233,150,122
🟩	🟩	darkseagreen	#8fbc8f	143,188,143
🟦	🟦	darkslateblue	#483d8b	72,61,139
⬛	⬛	darkslategray	#2f4f4f	47,79,79
	⬛	darkslategrey	#2f4f4f	47,79,79
🟦	🟦	darkturquoise	#00ced1	0,206,209
🟪	🟪	darkviolet	#9400d3	148,0,211
🟥	🟥	deeppink	#ff1493	255,20,147
🟦	🟦	deepskyblue	#00bfff	0,191,255
⬛	⬛	dimgray	#696969	105,105,105
	⬛	dimgrey	#696969	105,105,105
🟦	🟦	dodgerblue	#1e90ff	30,144,255
🟥	🟥	firebrick	#b22222	178,34,34
⬜	⬜	floralwhite	#fffaf0	255,250,240
🟩	🟩	forestgreen	#228b22	34,139,34
🟪	🟪	fuchsia	#ff00ff	255,0,255
⬜	⬜	gainsboro	#dcdcdc	220,220,220
⬜	⬜	ghostwhite	#f8f8ff	248,248,255

■	■			
■	■	gold	#ffd700	255,215,0
■	■	goldenrod	#daa520	218,165,32
■	■	gray	#808080	128,128,128
■	■	green	#008000	0,128,0
■	■	greenyellow	#adff2f	173,255,47
	■	grey	#808080	128,128,128
■	■	honeydew	#f0fff0	240,255,240
■	■	hotpink	#ff69b4	255,105,180
■	■	indianred	#cd5c5c	205,92,92
■	■	indigo	#4b0082	75,0,130
■	■	ivory	#fffff0	255,255,240
■	■	khaki	#f0e68c	240,230,140
■	■	lavender	#e6e6fa	230,230,250
■	■	lavenderblush	#fff0f5	255,240,245
■	■	lawngreen	#7cfc00	124,252,0
■	■	lemonchiffon	#fffacd	255,250,205
■	■	lightblue	#add8e6	173,216,230
■	■	lightcoral	#f08080	240,128,128
■	■	lightcyan	#e0ffff	224,255,255
■	■	lightgoldenrodyellow	#fafad2	250,250,210
	■	lightgray	#d3d3d3	211,211,211
■	■	lightgreen	#90ee90	144,238,144

■	■	lightgrey	#d3d3d3	211,211,211
■	■	lightpink	#ffb6c1	255,182,193
■	■	lightsalmon	#ffa07a	255,160,122
■	■	lightseagreen	#20b2aa	32,178,170
■	■	lightskyblue	#87cefa	135,206,250
■	■	lightslategray	#778899	119,136,153
	■	lightslategrey	#778899	119,136,153
■	■	lightsteelblue	#b0c4de	176,196,222
■	■	lightyellow	#ffffe0	255,255,224
■	■	lime	#00ff00	0,255,0
■	■	limegreen	#32cd32	50,205,50
■	■	linen	#faf0e6	250,240,230
■	■	magenta	#ff00ff	255,0,255
■	■	maroon	#800000	128,0,0
■	■	mediumaquamarine	#66cdaa	102,205,170
■	■	mediumblue	#0000cd	0,0,205
■	■	mediumorchid	#ba55d3	186,85,211
■	■	mediumpurple	#9370db	147,112,219
■	■	mediumseagreen	#3cb371	60,179,113
■	■	mediumslateblue	#7b68ee	123,104,238
■	■	mediumspringgreen	#00fa9a	0,250,154
■	■	mediumturquoise	#48d1cc	72,209,204

🟪	🟪	mediumvioletred	#c71585	199,21,133
🟦	🟦	midnightblue	#191970	25,25,112
⬜	⬜	mintcream	#f5fffa	245,255,250
⬜	⬜	mistyrose	#ffe4e1	255,228,225
🟨	🟨	moccasin	#ffe4b5	255,228,181
🟨	🟨	navajowhite	#ffdead	255,222,173
🟦	🟦	navy	#000080	0,0,128
⬜	⬜	oldlace	#fdf5e6	253,245,230
🟫	🟫	olive	#808000	128,128,0
🟩	🟩	olivedrab	#6b8e23	107,142,35
🟧	🟧	orange	#ffa500	255,165,0
🟥	🟥	orangered	#ff4500	255,69,0
🟪	🟪	orchid	#da70d6	218,112,214
🟨	🟨	palegoldenrod	#eee8aa	238,232,170
🟩	🟩	palegreen	#98fb98	152,251,152
🟦	🟦	paleturquoise	#afeeee	175,238,238
🟥	🟥	palevioletred	#db7093	219,112,147
⬜	⬜	papayawhip	#ffefd5	255,239,213
🟧	🟧	peachpuff	#ffdab9	255,218,185
🟫	🟫	peru	#cd853f	205,133,63
🟥	🟥	pink	#ffc0cb	255,192,203
🟪	🟪	plum	#dda0dd	221,160,221

🟦	🟦	powderblue	#b0e0e6	176,224,230
🟪	🟪	purple	#800080	128,0,128
🟥	🟥	red	#ff0000	255,0,0
🟫	🟫	rosybrown	#bc8f8f	188,143,143
🟦	🟦	royalblue	#4169e1	65,105,225
🟫	🟫	saddlebrown	#8b4513	139,69,19
🟥	🟥	salmon	#fa8072	250,128,114
🟧	🟧	sandybrown	#f4a460	244,164,96
🟩	🟩	seagreen	#2e8b57	46,139,87
⬜	⬜	seashell	#fff5ee	255,245,238
🟫	🟫	sienna	#a0522d	160,82,45
⬜	⬜	silver	#c0c0c0	192,192,192
🟦	🟦	skyblue	#87ceeb	135,206,235
🟪	🟪	slateblue	#6a5acd	106,90,205
⬛	⬛	slategray	#708090	112,128,144
	⬛	slategrey	#708090	112,128,144
⬜	⬜	snow	#fffafa	255,250,250
🟩	🟩	springgreen	#00ff7f	0,255,127
🟦	🟦	steelblue	#4682b4	70,130,180
🟫	🟫	tan	#d2b48c	210,180,140
🟦	🟦	teal	#008080	0,128,128
🟪	🟪	thistle	#d8bfd8	216,191,216
🟥	🟥	tomato	#ff6347	255,99,71

🟥	🟥			
🟦	🟦	turquoise	#40e0d0	64,224,208
🟪	🟪	violet	#ee82ee	238,130,238
🟨	🟨	wheat	#f5deb3	245,222,179
		white	#ffffff	255,255,255
⬜	⬜	whitesmoke	#f5f5f5	245,245,245
🟨	🟨	yellow	#ffff00	255,255,0
🟩	🟩	yellowgreen	#9acd32	154,205,50

How to Create a Simple CSS Stylesheet Using Notepad

PARTS

OTHER SECTIONS

Written by Jack Lloyd

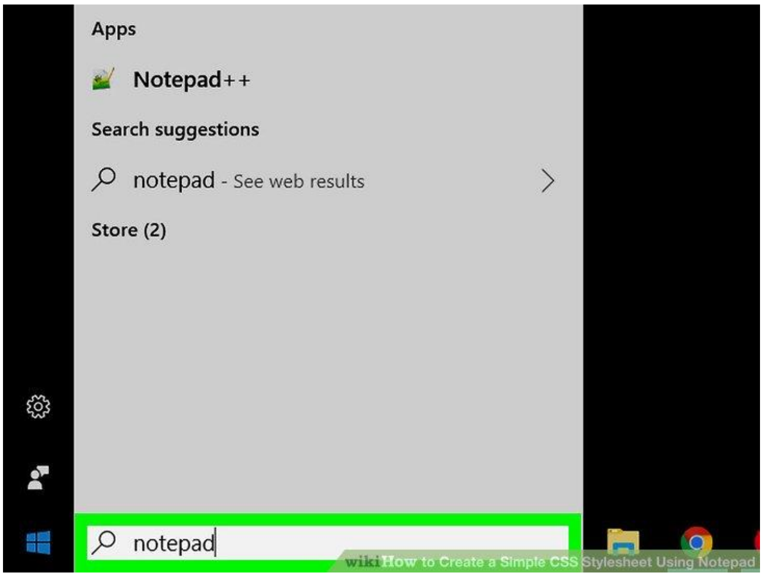
Last Updated: February 2, 2024

This wikiHow teaches you how to use Windows' Notepad app to store information for a webpage written in HTML and CSS. HTML is the programming language used to create your webpage, while CSS is the language that determines the style—color, font, and so on—of the HTML elements on the webpage.

Part1

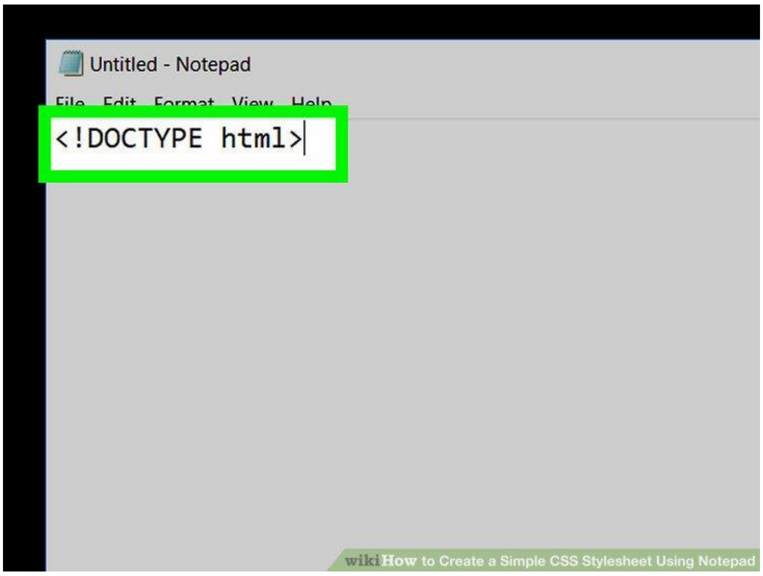
Creating an HTML Page

1


Apps
Notepad++
Search suggestions
notepad - See web results
Store (2)
notepad
wikiHow to Create a Simple CSS Stylesheet Using Notepad

Open Notepad. Open **Start** ⊞, type in `notepad`, and click the blue **Notepad** app at the top of the Start window.

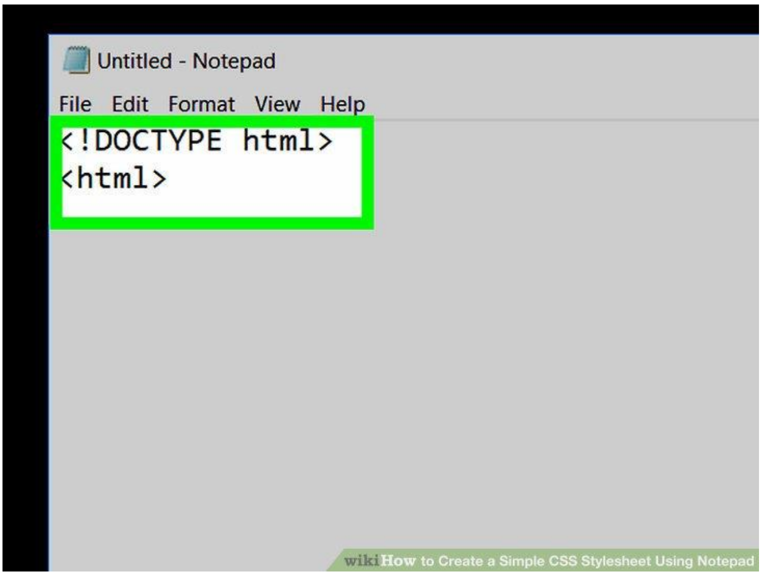
2


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple CSS Stylesheet Using Notepad

```
<!DOCTYPE html>
```

Indicate the document type. Type `<!DOCTYPE html>` into Notepad, then press ↲ Enter to start a new line.

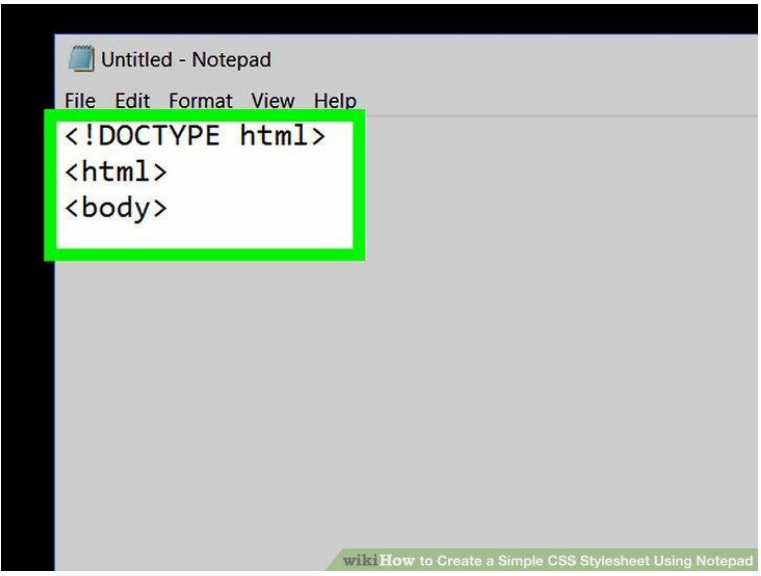
3


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple CSS Stylesheet Using Notepad

```
<!DOCTYPE html>
<html>
```

Add the HTML tag. Type in `<html>` and press ↲ Enter.

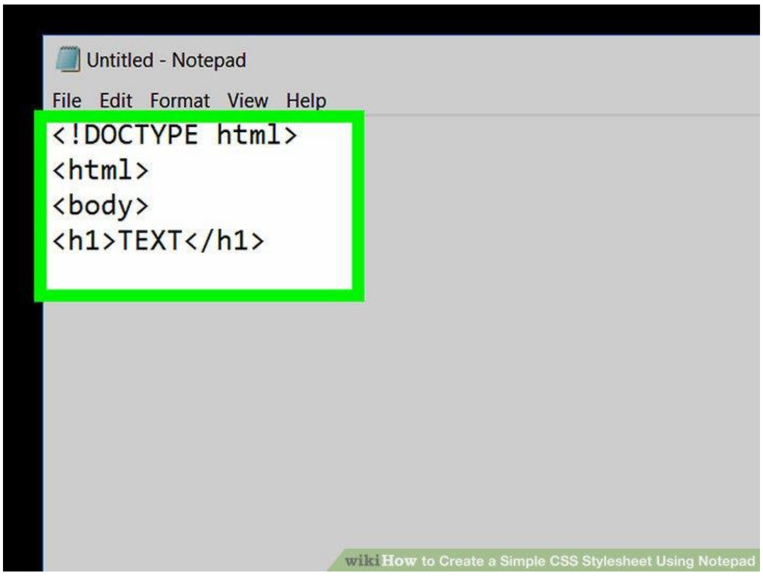
4


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple CSS Stylesheet Using Notepad

```
<!DOCTYPE html>
<html>
<body>
```

Enter the BODY tag. Type in `<body>` and press ↲ Enter. You can now begin entering your webpage's information.

5

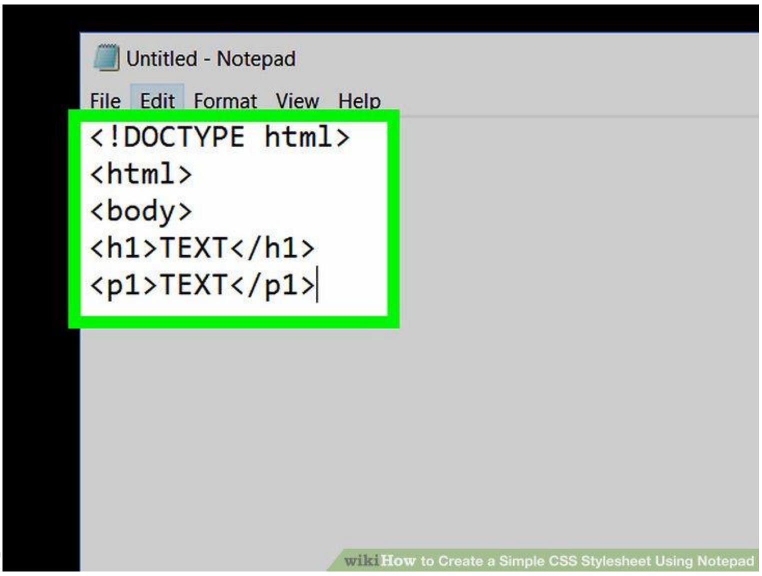

Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple CSS Stylesheet Using Notepad

```
<!DOCTYPE html>
<html>
<body>
<h1>TEXT</h1>
```

Add a header. Type in `<h1>TEXT</h1>`, making sure to replace "TEXT" with your preferred page heading, and press ↲ Enter.

- For example, to create a page header that says "Welcome!", you would type `<h1>Welcome!</h1>` into Notepad.

6

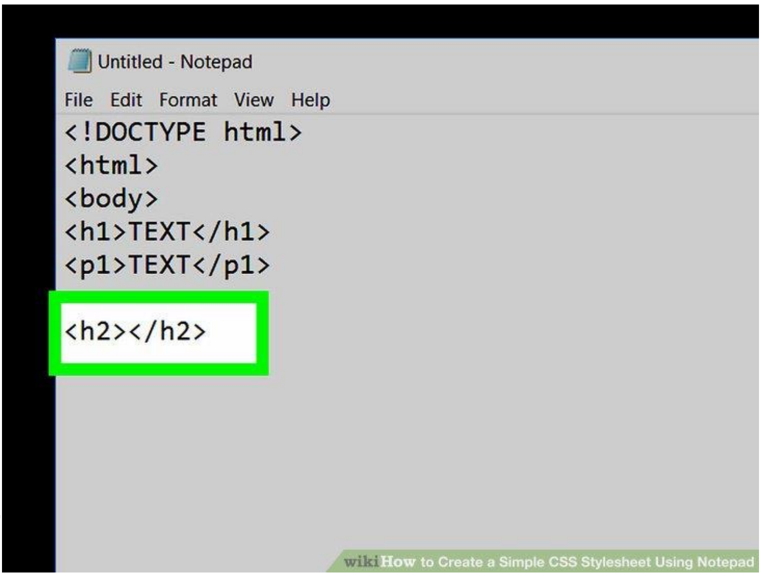

Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple CSS Stylesheet Using Notepad

```
<!DOCTYPE html>
<html>
<body>
<h1>TEXT</h1>
<p1>TEXT</p1>
```

Add text below the header. Type in `<p1>TEXT</p1>`, making sure to replace "TEXT" with your preferred message, and press ↲ Enter.

- For example, to add text that says "I am an iguana", you would enter `<p1>I am an iguana</p1>` into Notepad.

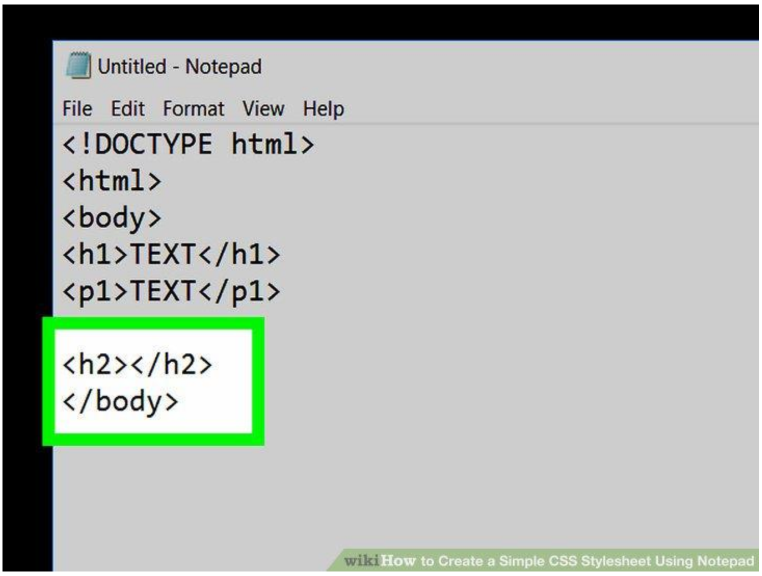
7


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple CSS Stylesheet Using Notepad

```
<!DOCTYPE html>
<html>
<body>
<h1>TEXT</h1>
<p1>TEXT</p1>

<h2></h2>
```

Add more headers and paragraphs. Each subsequent header and paragraph must have an ascending number applied to it; for example, your second header will have `<h2></h2>` tags around it, and the second paragraph will have `<p2></p2>` tags.

- Make sure that you're continuing to press ↲ Enter after each line of code.


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple CSS Stylesheet Using Notepad

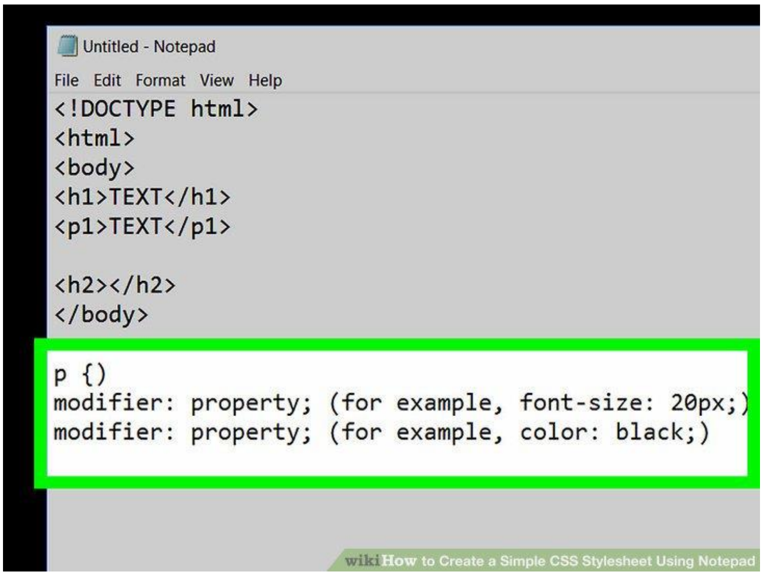
```
<!DOCTYPE html>
<html>
<body>
<h1>TEXT</h1>
<p1>TEXT</p1>

<h2></h2>
</body>
```

8

Close the BODY and HTML tags. Once you've entered your last line of code, type in `</body>` on its own line and press ↲ Enter, then type in `</html>`. Your document is now ready to be styled with CSS.

Part2

Adding CSS


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple CSS Stylesheet Using Notepad

```
<!DOCTYPE html>
<html>
<body>
<h1>TEXT</h1>
<p1>TEXT</p1>

<h2></h2>
</body>

p {)
modifier: property; (for example, font-size: 20px;)
modifier: property; (for example, color: black;)
```

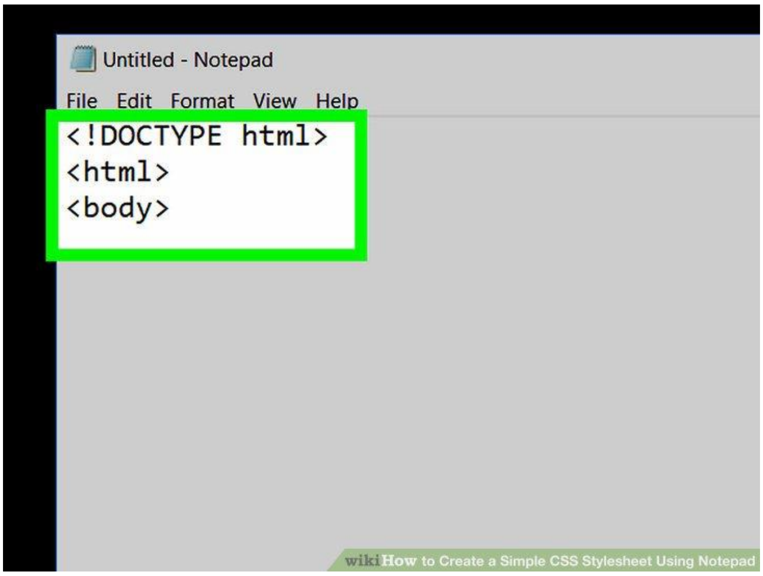
1

Understand how CSS works. You use CSS to change the appearance of an HTML element (e.g., a paragraph). CSS is typically written in the following line-by-line format:[1]

- `element tag {` (for example, `p {`)

- `modifier: property;` (for example, `font-size: 20px;`)
- `modifier: property;` (for example, `color: black;`)
- `}`

2

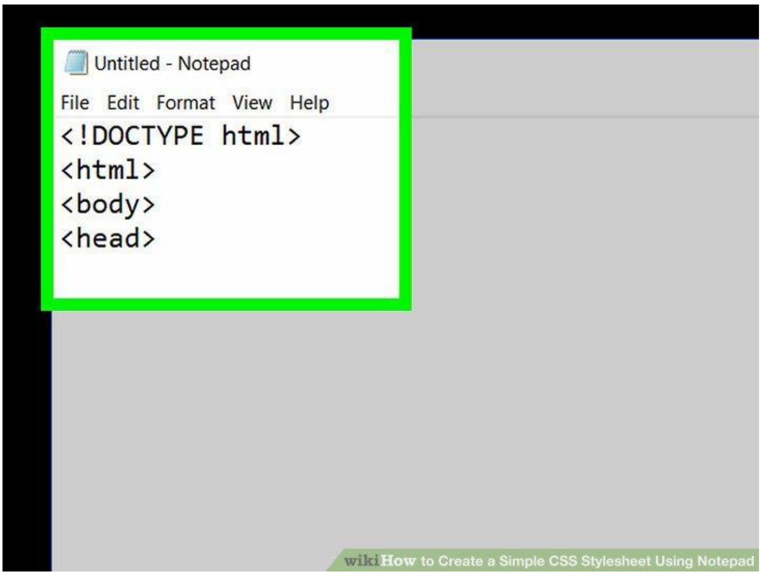


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple CSS Stylesheet Using Notepad

```
<!DOCTYPE html>
<html>
<body>
```

Place a space between the `<html>` and `<body>` tags. These should be near the top of the page.

3

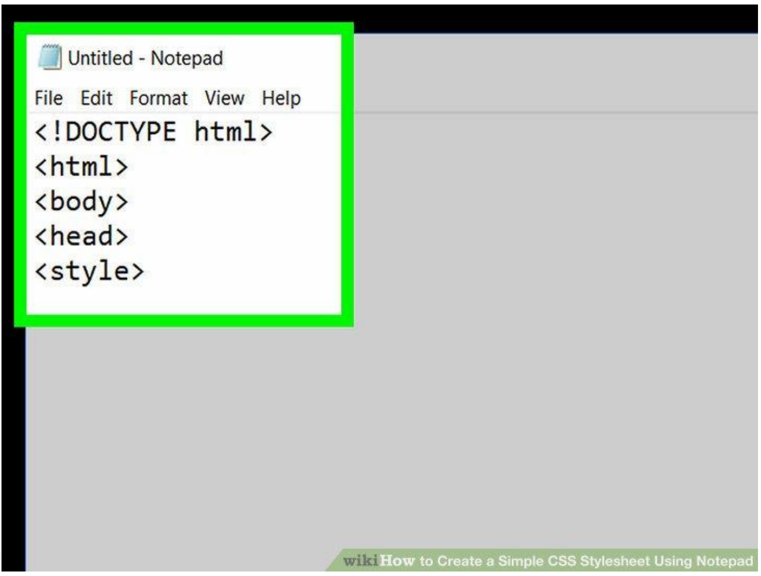


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple CSS Stylesheet Using Notepad

```
<!DOCTYPE html>
<html>
<body>
<head>
```

Enter a HEAD tag. Type in `<head>` and press ↲ Enter.

4

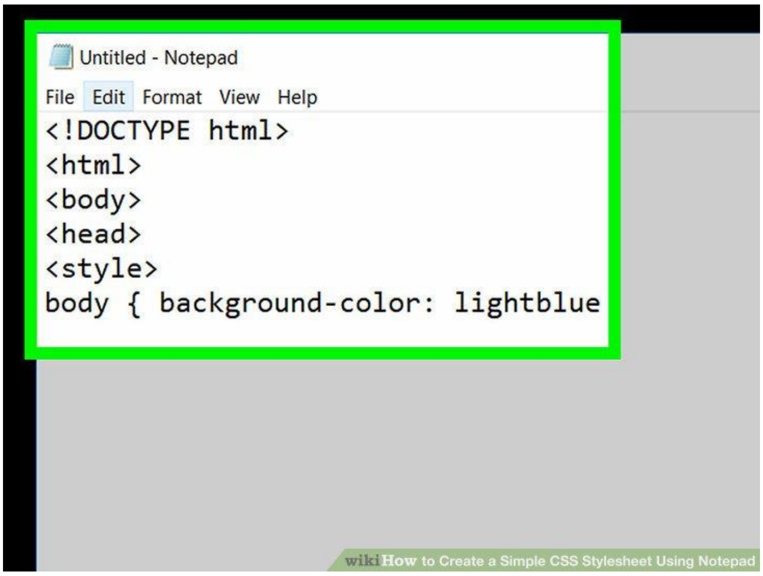


```
<!DOCTYPE html>
<html>
<body>
<head>
<style>
```

wikiHow to Create a Simple CSS Stylesheet Using Notepad

Add a STYLE tag. Type in `<style>` and press ↲ Enter.

5



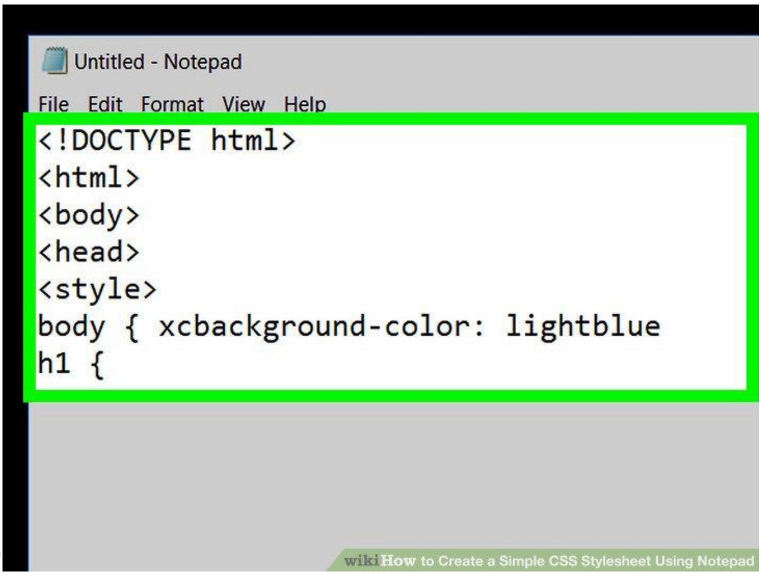
```
<!DOCTYPE html>
<html>
<body>
<head>
<style>
body { background-color: lightblue
```

wikiHow to Create a Simple CSS Stylesheet Using Notepad

Change your webpage's background color. To do so:

- Type in `body {` and press ↲ Enter.
- Type in `background-color: lightblue;` and press ↲ Enter.
    - You can use any supported color here, as well as "light" or "dark" modifiers.
- Type in } and press ↲ Enter.

6

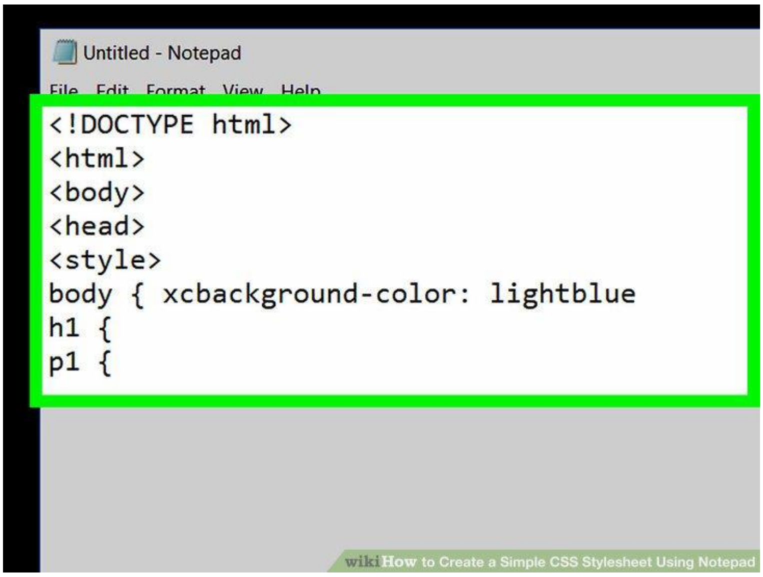

Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple CSS Stylesheet Using Notepad

```
<!DOCTYPE html>
<html>
<body>
<head>
<style>
body { xcbackground-color: lightblue
h1 {
```

Style your first header. Type in `h1 {` and press ↵ Enter, add a modifier and press ↵ Enter, and type in } and press ↵ Enter. You can add multiple modifiers to one element as long as each modifier is on its own line. Common modifiers include the following:

- **Text size** - Type in `font-size: 30px;` to set your text as 30-point font. Substitute any number that you want to use.
- **Text color** - Type in `color: black;` to make black text. Substitute any color that you want to use.
- **Text alignment** - Type in `text-align: center;` to center the text. You can also type `left` or `right` to align the text appropriately.
- **Font used** - Type in `font-family: times new roman;` to set your font as Times New Roman. You can also use fonts like `verdana` or `georgia`.

7

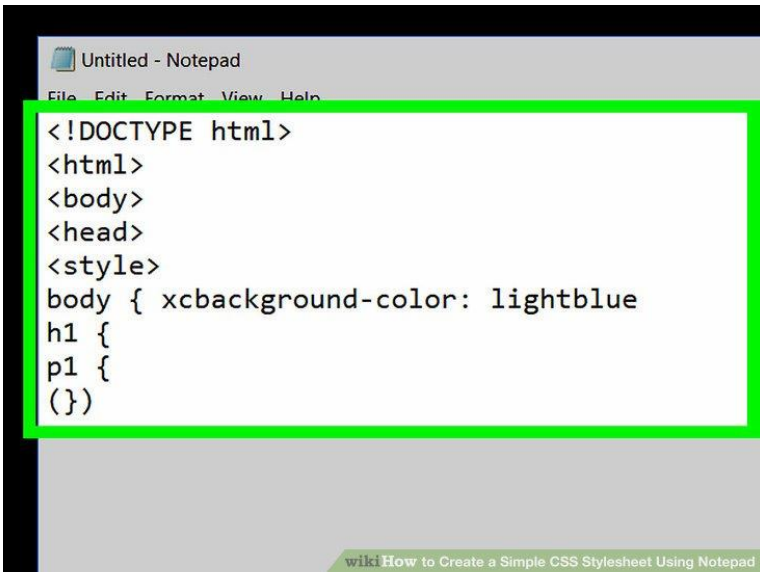

Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple CSS Stylesheet Using Notepad

```
<!DOCTYPE html>
<html>
<body>
<head>
<style>
body { xcbackground-color: lightblue
h1 {
p1 {
```

Style your first paragraph. Type in `p1 {` and press ↲ Enter, add a modifier and press ↲ Enter, and type in } and press ↲ Enter.

- The modifiers that you can use here are identical to the ones used for the header.

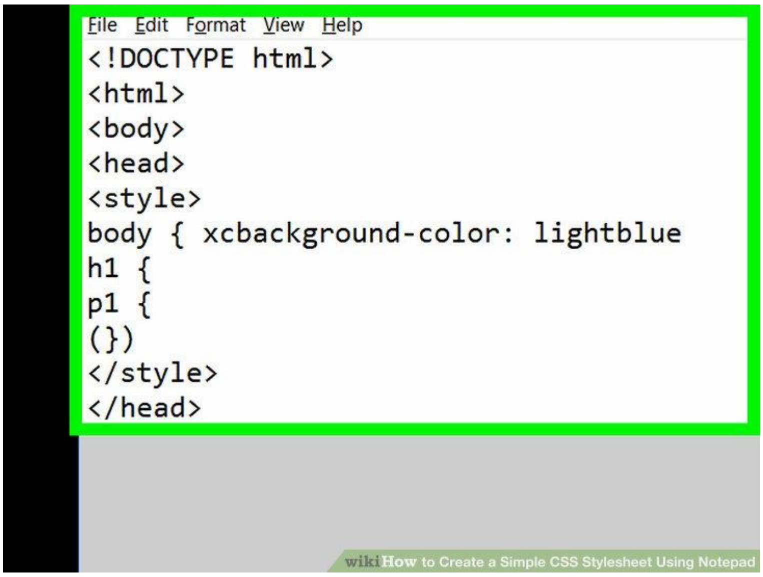
8


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple CSS Stylesheet Using Notepad

```
<!DOCTYPE html>
<html>
<body>
<head>
<style>
body { xcbackground-color: lightblue
h1 {
p1 {
(})
```

Style the rest of your document. You can style any element by referencing its tag text and placing an open curly bracket ({), adding modifiers, and then closing the bracket (}).

9

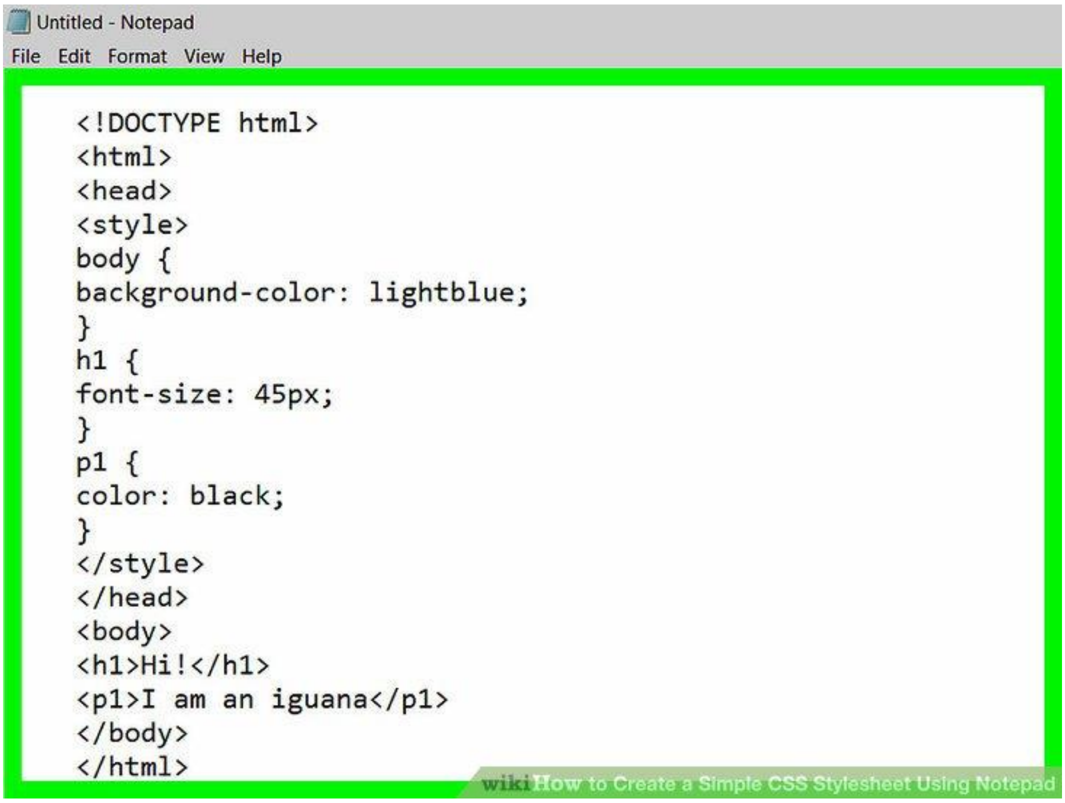

File Edit Format View Help
wikiHow to Create a Simple CSS Stylesheet Using Notepad

```
<!DOCTYPE html>
<html>
<body>
<head>
<style>
body { xcbackground-color: lightblue
h1 {
p1 {
(})
</style>
</head>
```

Close the STYLE and HEAD tags. On a new line below your last styling text, type in `</style>` and press ↲ Enter, then type in `</head>` and press ↲ Enter. Your CSS sheet is complete, meaning that you can now review and save it.

Part3

Saving the Document


Untitled - Notepad
File Edit Format View Help
wikiHow to Create a Simple CSS Stylesheet Using Notepad

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
background-color: lightblue;
}
h1 {
font-size: 45px;
}
p1 {
color: black;
}
</style>
</head>
<body>
<h1>Hi!</h1>
<p1>I am an iguana</p1>
</body>
</html>
```

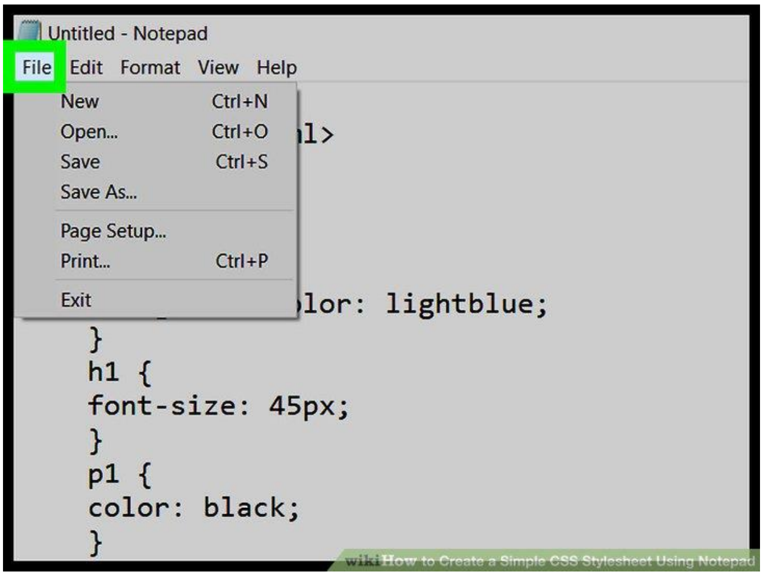
1

Review your CSS stylesheet. Your CSS document will vary slightly, but it should look something like this:

- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `body {`
- `background-color: lightblue;`
- `}`
- `h1 {`
- `font-size: 45px;`
- `}`
- `p1 {`
- `color: black;`
- `}`
- `</style>`

- </head>
- <body>
- <h1>Hi!</h1>
- <p1>I am an iguana</p1>
- </body>
- </html>

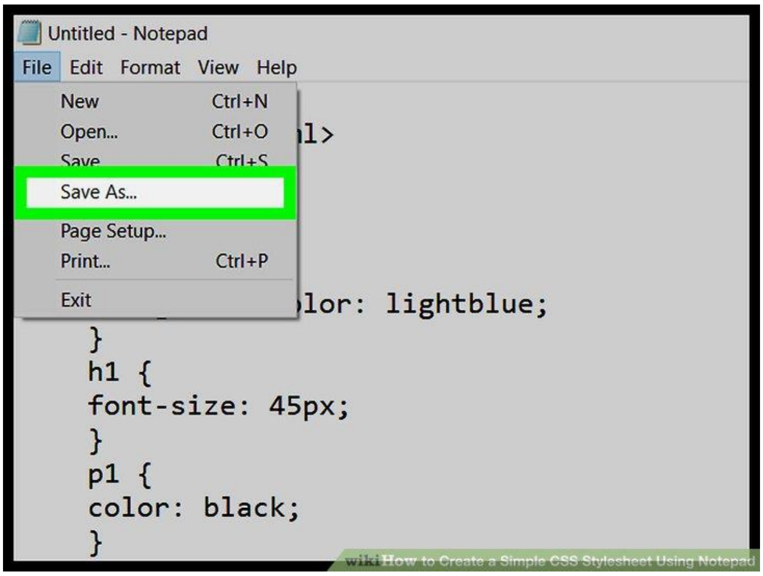
2



```
l>
lor: lightblue;
}
h1 {
font-size: 45px;
}
p1 {
color: black;
}
```

Click File**.** It's in the top-left corner of the Notepad window. A drop-down menu will appear.

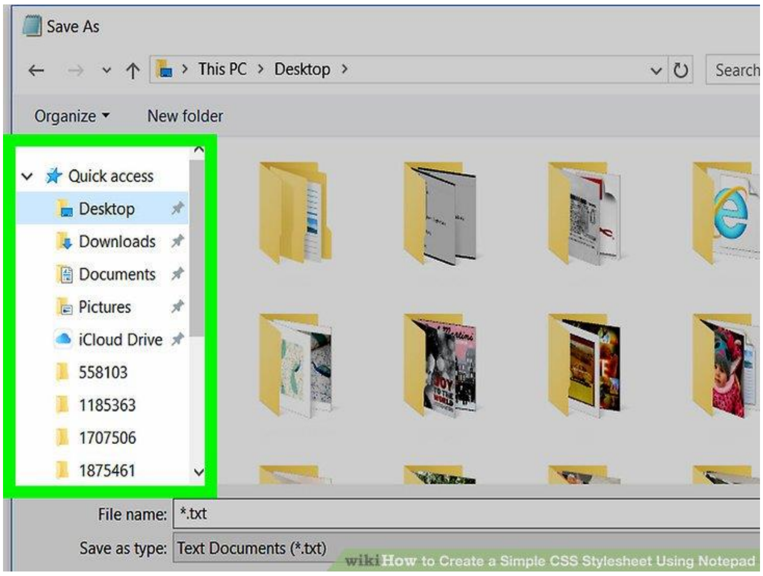
3



```
l>
lor: lightblue;
}
h1 {
font-size: 45px;
}
p1 {
color: black;
}
```

Click Save As…**.** You'll see this near the bottom of the drop-down menu. Clicking it prompts a window to open.

4

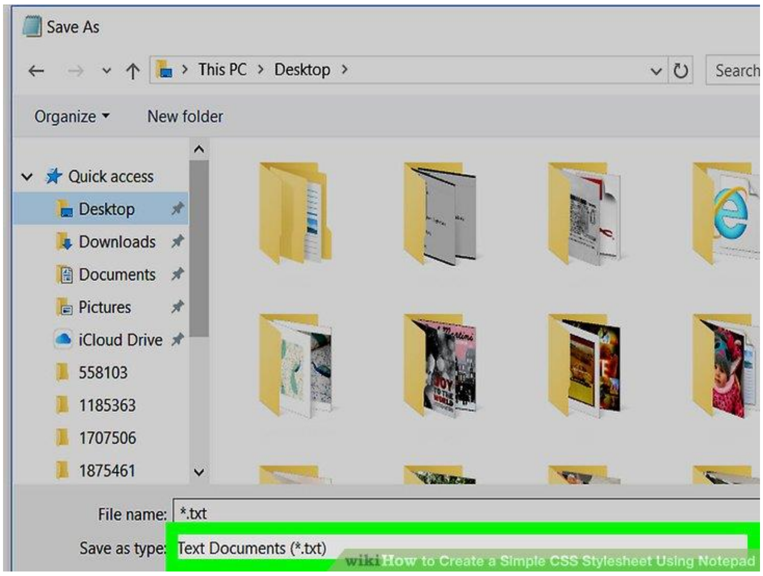


File name: *.txt

Save as type: Text Documents (*.txt)

Select a save location. Click a folder (e.g., **Desktop**) on the left side of the window.

5

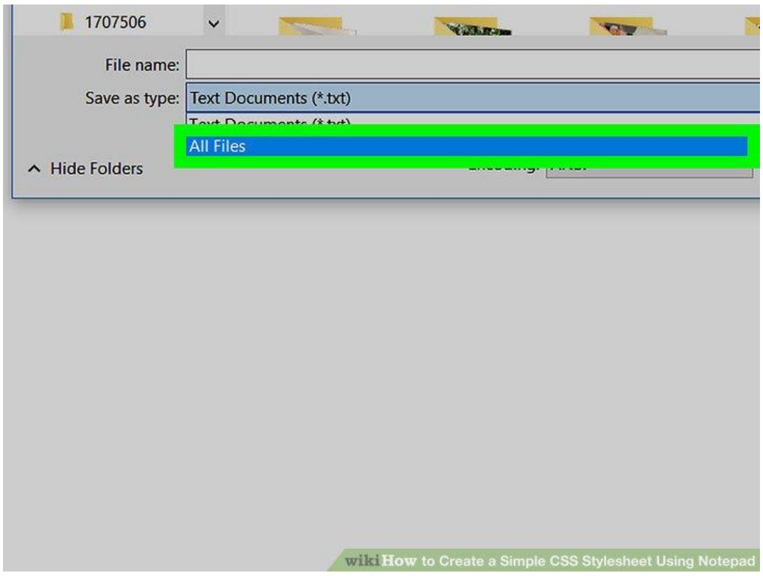


File name: *.txt

Save as type: Text Documents (*.txt)

Click the "Save as type" drop-down box. A drop-down menu will appear.

6

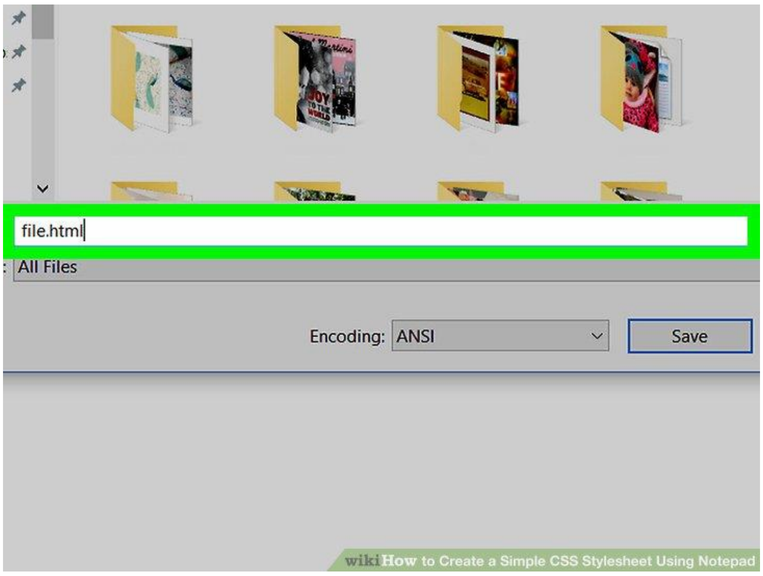


File name:

Save as type: Text Documents (*.txt)

Click All Files. It's in the drop-down menu.

7



file.html

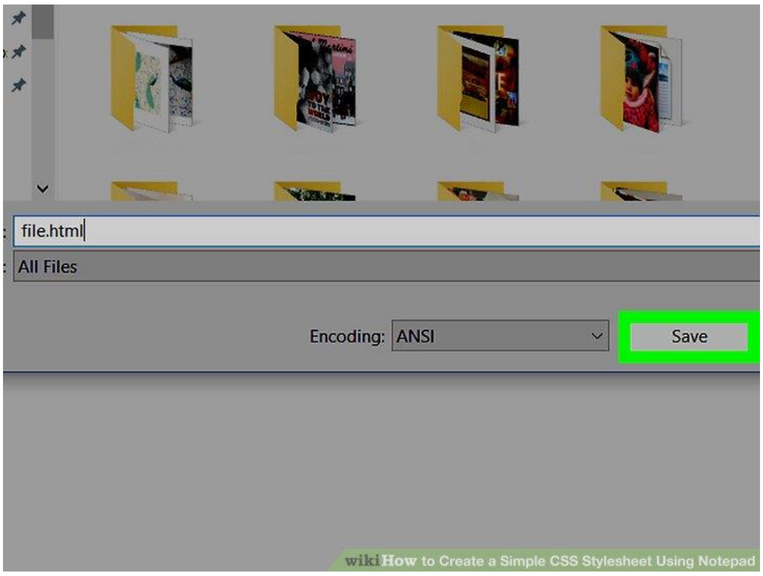
All Files

Encoding: ANSI

Name your file with a ".html" extension. In the "File name" text field, type in your preferred document name (e.g., "My CSS") followed by `.html`.

- For example, if you named your file "My CSS", you'd type `my css.html` into the field.
- If you use a program that can run ".css" files, you can use `.css` instead of `.html` here.

8


file.html

All Files

Encoding: ANSI

wikiHow to Create a Simple CSS Stylesheet Using Notepad

Click Save. It's in the bottom-right side of the "Save As" window. This will save your CSS sheet in an executable format, meaning that you can open it in your preferred browser or HTML editor rather than in Notepad.

Community Q&A

- Question

What is the procedure to compile using a CSS file?

Community Answer

CSS files do not need to be compiled. You may just link it to your .html header.

- Question

How do I create an account?

Community Answer

Notepad files will be saved to your computer itself, so no account is necessary.

- Question

How do I run the CSS file?

Community Answer

CSS files are stylesheets that extend HTML pages with styles. They are not able to be run, but must be specified as links in a tag.

- **Question**

Where will you save the CSS file?

CageyCat

Top Answerer

You should make a folder called My Webs on your computer under My Documents. Make a Folder in My Webs that matches your Account Name, such as hippos. Under hippos (folder), make sub folders for your website, such as images. Then, put your css (file extension) file loose (no folder) under hippos. Same with all your html (file extension) files for your webpages, except if you want to sort them by subjects in sub folders. Being organized from the beginning will make it much easier as your site grows.

- **Question**

After having my webpage source code, how do I now publish? Do I need to purchase anything?

CageyCat

Top Answerer

Filezilla is a free program for FTP (File Transfer Protocol). You'll need pieces of info from your Host to set up Filezilla.

Tips

- When coding in HTML or CSS, it doesn't really matter how many spaces are between lines of code; you can press ↵ Enter several times after each line without changing the function of the program.
- Try indenting different parts of your CSS stylesheet to make finding elements easier. For example, you might indent header code once and paragraph code twice.

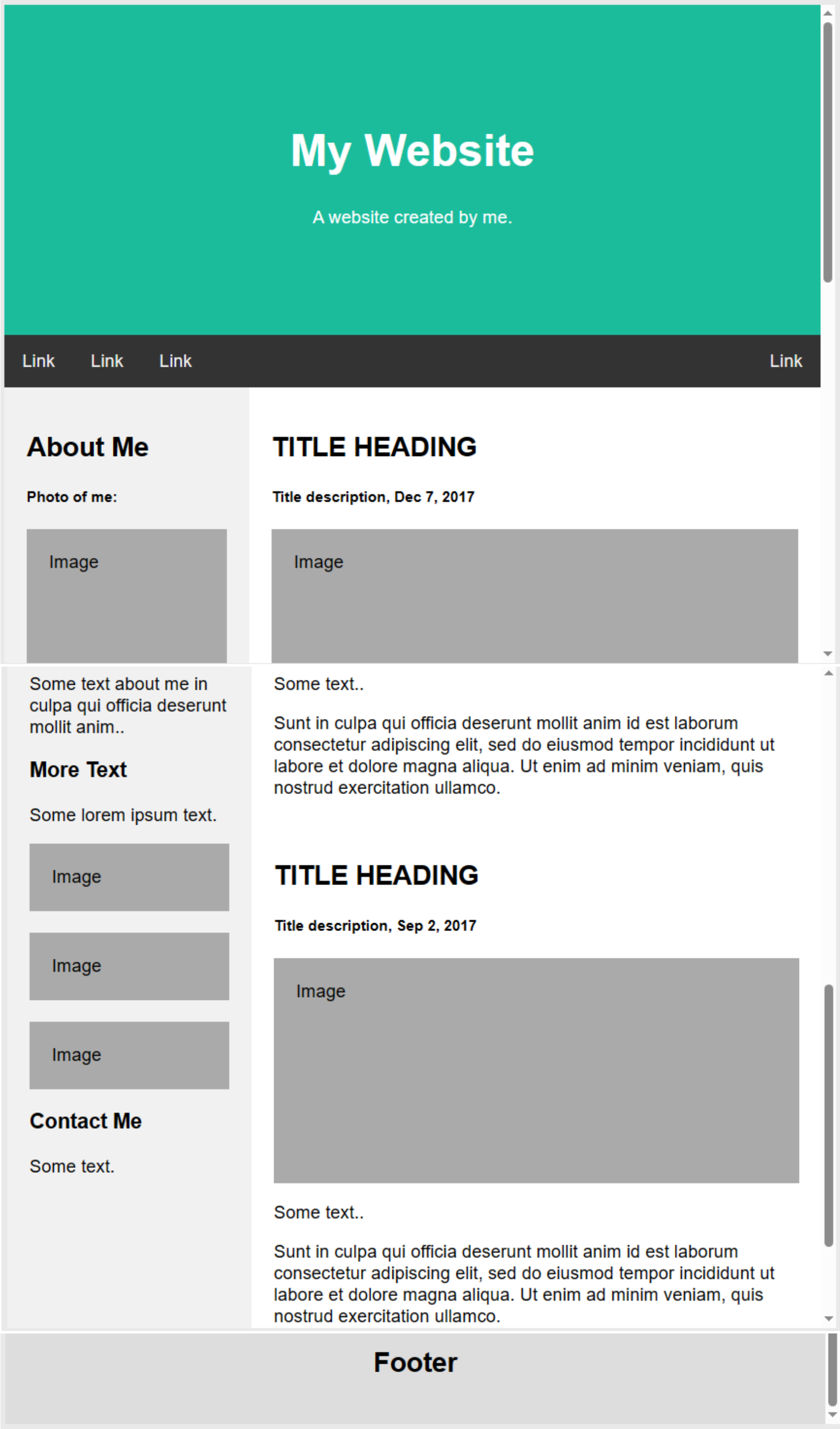
Warnings

- Always test your code before uploading it to a website or sharing it with other people.

How TO - Make a Website

Learn how to create a responsive website that will work on all devices, PC, laptop, tablet, and phone.

Create a Website from Scratch



My Website

A website created by me.

About Me

Photo of me:

Some text about me in culpa qui officia deserunt mollit anim..

More Text

Some lorem ipsum text.

Contact Me

Some text.

TITLE HEADING

Title description, Dec 7, 2017

Some text..

Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.

TITLE HEADING

Title description, Sep 2, 2017

Some text..

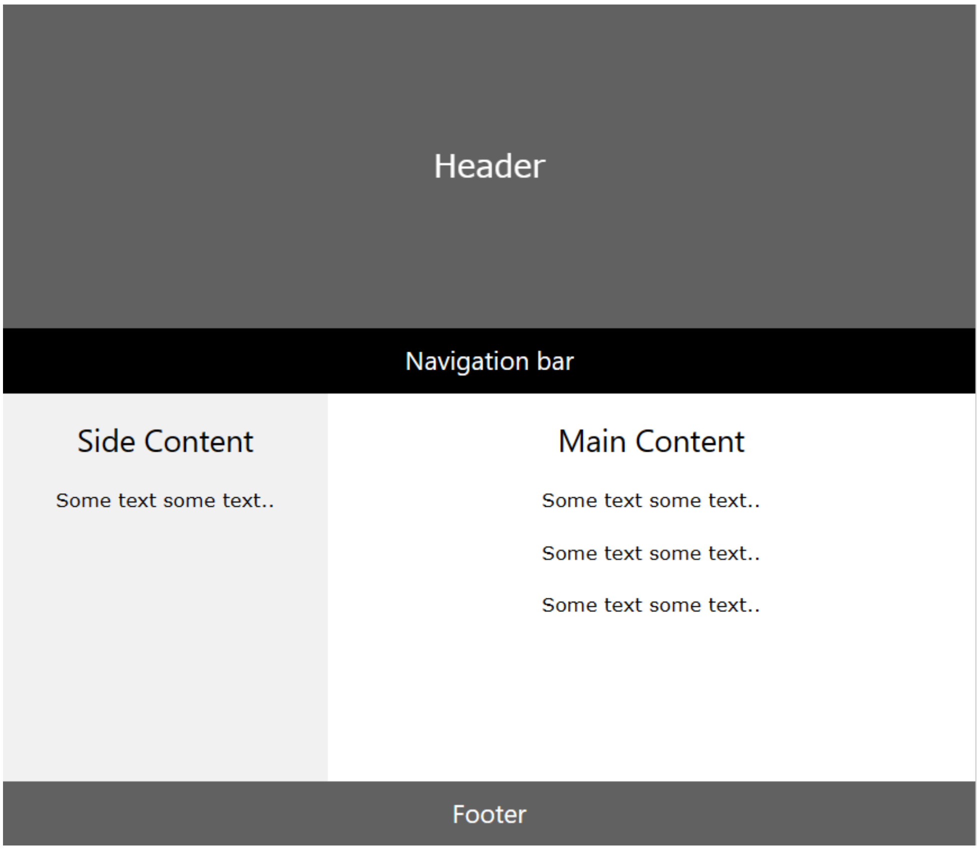
Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.

Footer

Demo Try it Yourself

A "Layout Draft"

It can be wise to draw a layout draft of the page design before creating a website:



First Step - Basic HTML Page

HTML is the standard markup language for creating websites and CSS is the language that describes the style of an HTML document. We will combine HTML and CSS to create a basic web page.

Note: If you don't know HTML and CSS, we suggest that you start by reading our HTML Tutorial.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Page Title</title>
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  font-family: Arial, Helvetica, sans-serif;
}
</style>
</head>
<body>

<h1>My Website</h1>
<p>A website created by me.</p>

</body>
</html>
```

Try it Yourself »

Example Explained

- The `<!DOCTYPE html>` declaration defines this document to be HTML5
- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about the document
- The `<title>` element specifies a title for the document
- The `<meta>` element should define the character set to be UTF-8
- The `<meta>` element with name="viewport" makes the website look good on all devices and screen resolutions
- The `<style>` element contains the styles for the website (layout/design)
- The `<body>` element contains the visible page content
- The `<h1>` element defines a large heading
- The `<p>` element defines a paragraph

Creating Page Content

Inside the `<body>` element of our website, we will use our "Layout Draft" and create:

- A header
- A navigation bar
- Main content
- Side content
- A footer

Header

A header is usually located at the top of the website (or right below a top navigation menu). It often contains a logo or the website name:

```
<div class="header">
  <h1>My Website</h1>
  <p>A website created by me.</p>
</div>
```

Then we use CSS to style the header:

```
.header {
  padding: 80px; /* some padding */
  text-align: center; /* center the text */
  background: #1abc9c; /* green background */
  color: white; /* white text color */
}

/* Increase the font size of the <h1> element */
.header h1 {
  font-size: 40px;
}
```

Try it Yourself »

Navigation Bar

A navigation bar contains a list of links to help visitors navigating through your website:

```
<div class="navbar">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#" class="right">Link</a>
</div>
```

Use CSS to style the navigation bar:

```
/* Style the top navigation bar */
.navbar {
  overflow: hidden; /* Hide overflow */
  background-color: #333; /* Dark background color */
}
```

```
/* Style the navigation bar links */
.navbar a {
  float: left; /* Make sure that the links stay side-by-side */
  display: block; /* Change the display to block, for responsive
reasons (see below) */
  color: white; /* White text color */
  text-align: center; /* Center the text */
  padding: 14px 20px; /* Add some padding */
  text-decoration: none; /* Remove underline */
}

/* Right-aligned link */
.navbar a.right {
  float: right; /* Float a link to the right */
}

/* Change color on hover/mouse-over */
.navbar a:hover {
  background-color: #ddd; /* Grey background color */
  color: black; /* Black text color */
}
```

Try it Yourself »

Content

Create a 2-column layout, divided into a "side content" and a "main content".

```
<div class="row">
  <div class="side">...</div>
  <div class="main">...</div>
</div>
```

We use CSS Flexbox to handle the layout:

```
/* Ensure proper sizing */
* {
  box-sizing: border-box;
}

/* Column container */
.row {
  display: flex;
  flex-wrap: wrap;
}
```

```
/* Create two unequal columns that sits next to each other */
/* Sidebar/left column */
.side {
  flex: 30%; /* Set the width of the sidebar */
  background-color: #f1f1f1; /* Grey background color */
  padding: 20px; /* Some padding */
}

/* Main column */
.main {
  flex: 70%; /* Set the width of the main content */
  background-color: white; /* White background color */
  padding: 20px; /* Some padding */
}
```

Try it Yourself »

Then add media queries to make the layout responsive. This will make sure that your website looks good on all devices (desktops, laptops, tablets and phones). Resize the browser window to see the result.

```
/* Responsive layout - when the screen is less than 700px wide, make
the two columns stack on top of each other instead of next to each
other */
@media screen and (max-width: 700px) {
  .row {
    flex-direction: column;
  }
}

/* Responsive layout - when the screen is less than 400px wide, make
the navigation links stack on top of each other instead of next to each
other */
@media screen and (max-width: 400px) {
  .navbar a {
    float: none;
    width: 100%;
  }
}
```

Try it Yourself »

Tip: To create a different kind of layout, just change the flex width (but make sure that it adds up to 100%).

Tip: Do you wonder how the @media rule works? Read more about it in our CSS Media Queries chapter.

Tip: To learn more about the Flexible Box Layout Module, read our CSS Flexbox chapter.

What is box-sizing?

You can easily create three floating boxes side by side. However, when you add something that enlarges the width of each box (e.g. padding or borders), the box will break. The `box-sizing` property allows us to include the padding and border in the box's total width (and height), making sure that the padding stays inside of the box and that it does not break.

You can read more about the box-sizing property in our CSS Box Sizing Tutorial.

Footer

At last, we will add a footer.

```
<div class="footer">
  <h2>Footer</h2>
</div>
```

And style it:

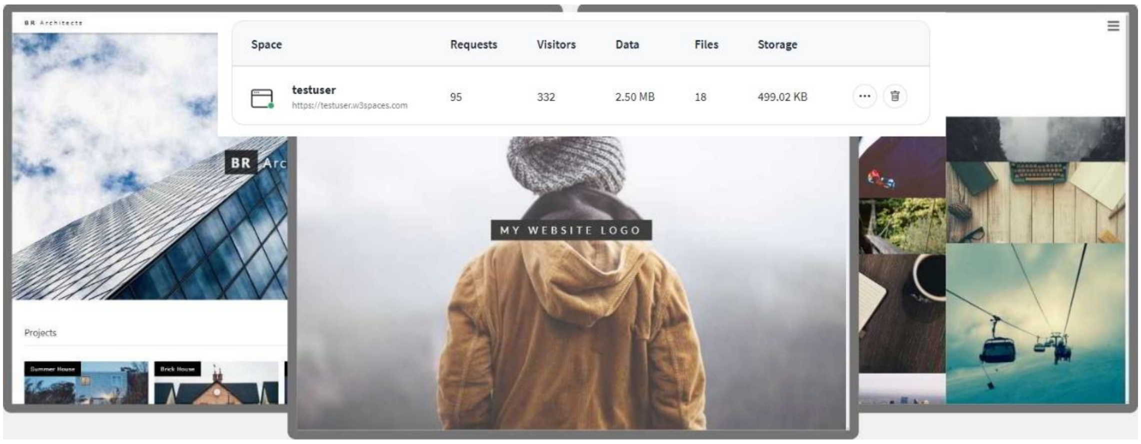
```
.footer {
  padding: 20px; /* Some padding */
  text-align: center; /* Center text*/
  background: #ddd; /* Grey background */
}
```

Try it Yourself »

Congratulations! You have built a responsive website from scratch.

W3Schools Spaces

If you want to create your own website and host your .html files, try our **website builder**, called **W3schools Spaces**:



Space	Requests	Visitors	Data	Files	Storage
testuser https://testuser.w3spaces.com	95	332	2.50 MB	18	499.02 KB

Get your own Website

How TO - Create a Free Website

Build and host your website with W3Schools Spaces.

Get started with your free website in a few clicks.

Everything you need right in the browser.

It is easy to use - try it!

Get started for free »

Create Your First Website with W3Schools Spaces

W3Schools Spaces is a personal place where you can build and experiment with code and host your own website.

With W3Schools Spaces you can build with HTML, CSS and JavaScript.

Edit code directly in your browser.

Upload and host your files and images.

Start to grow your online presence today!

Why Build with W3Schools Spaces?

Spaces is made by web developers for web developers.

1. The interface is simple and easy to use.
2. Host and edit your files right in the browser.
3. Free templates.

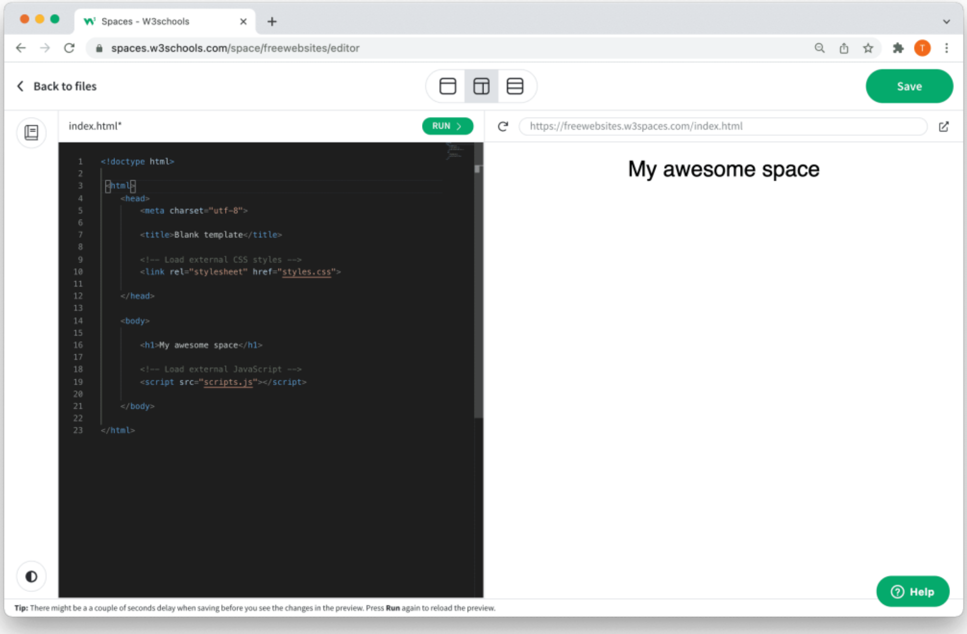
It is **free** to get started and you do not have to enter your credit card.

Edit and Preview Code

The editor is easy to use - which helps you to focus on the most important things.

Check the responsiveness of your site with the different preview alternatives.

Preview changes on your site, live!



Spaces - W3schools
spaces.w3schools.com/space/freewebsites/editor
Back to files
Save
index.html*
RUN
https://freewebsites.w3spaces.com/index.html
My awesome space
Help

```
<!doctype html>

<html>
    <head>
        <meta charset="utf-8">

        <title>Blank template</title>

        <!-- Load external CSS styles -->
        <link rel="stylesheet" href="styles.css">

    </head>

    <body>

        <h1>My awesome space</h1>

        <!-- Load external JavaScript -->
        <script src="scripts.js"></script>

    </body>

</html>
```

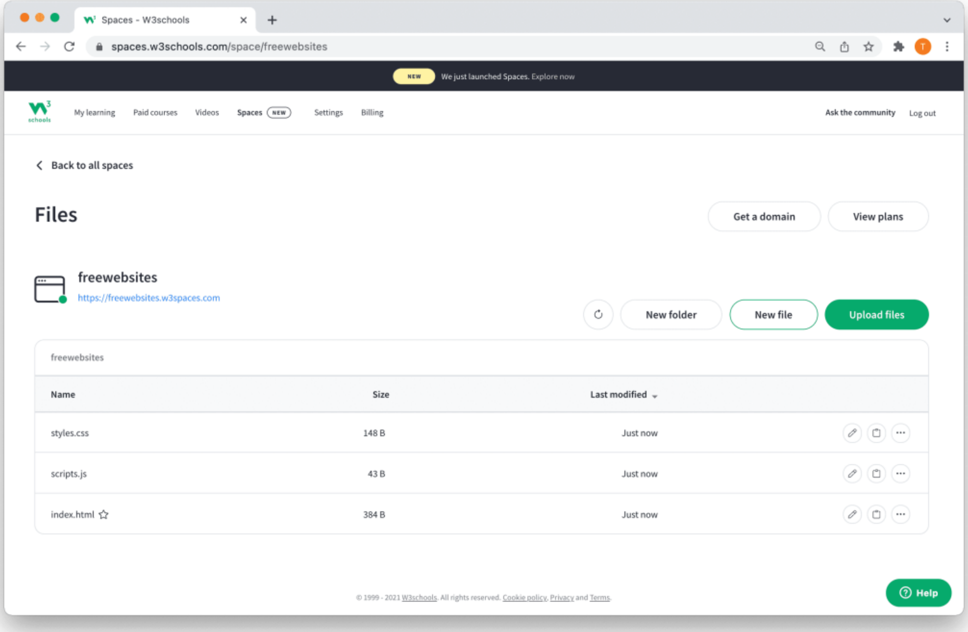
Tip: There might be a a couple of seconds delay when saving before you see the changes in the preview. Press **Run** again to reload the preview.

Build From Anywhere

Host your files and images in the cloud.

Keep organized by creating a structure with folders.

Everything you need right in the browser.



freewebsites

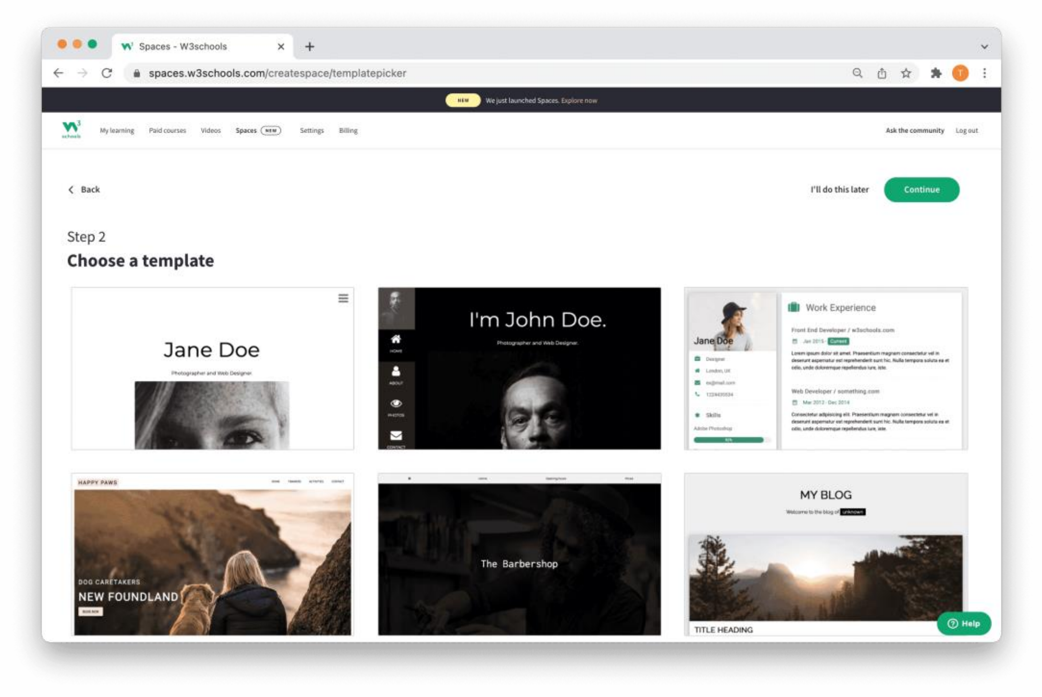
Name	Size	Last modified
styles.css	148 B	Just now
scripts.js	43 B	Just now
index.html ☆	384 B	Just now

© 1999 - 2021 W3schools. All rights reserved. Cookie policy, Privacy and Terms.

Free Templates

Browse and use our responsive website templates.

Modify, save, share, and use them in your projects.



The templates are powered by **W3.CSS**

What Do I Need to Know to Get Started?

HTML, CSS and JavaScript are the foundational languages to build a website.

1. **Create the structure with HTML.** The first thing you have to learn, is HTML, which is the standard mark-up language for creating web pages.

   Learn HTML »

2. **Style with CSS.** The next step is to learn CSS, to set the layout of your web page with beautiful colours, fonts, and much more.

   Learn CSS »

3. **Make it interactive with JavaScript.** After studying HTML and CSS, you should learn JavaScript to create dynamic and interactive web pages for your users.

Learn JavaScript »

Do not worry if you do not know how to code. The most important thing is to get hands on, early on. Learning how to code is best done with getting practical. Start to build something today!

Let’s Get Started in a Few Steps

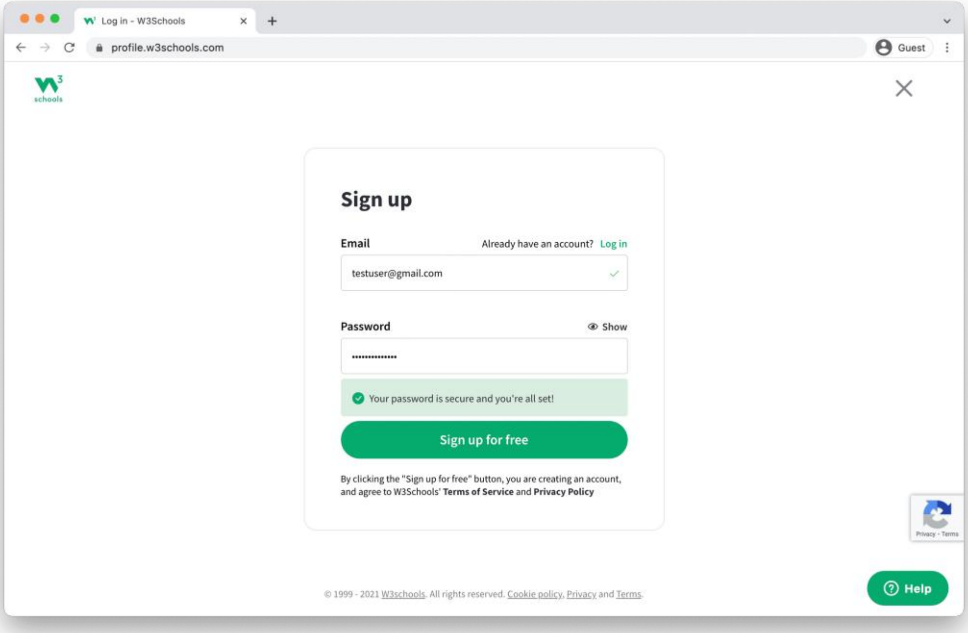
Do you already have a W3Schools Account? If so, skip the first step

Step One: Sign Up For an Account

To be able to use Spaces you need to sign up and get your account.

Let’s get you set up!

Go to W3Schools Profile - Click "**Sign up**" and enter your email and password, then click the "**Sign up for free**" button.



Sign up

Email: testuser@gmail.com

Already have an account? Log in

Password: ••••••••••••••

Your password is secure and you're all set!

By clicking the "Sign up for free" button, you are creating an account, and agree to W3Schools' **Terms of Service** and **Privacy Policy**

© 1999 - 2021 W3schools. All rights reserved. Cookie policy, Privacy and Terms.

Remember to **validate** your account in your email. Check the spam filter if you cannot find the validation email in your inbox.

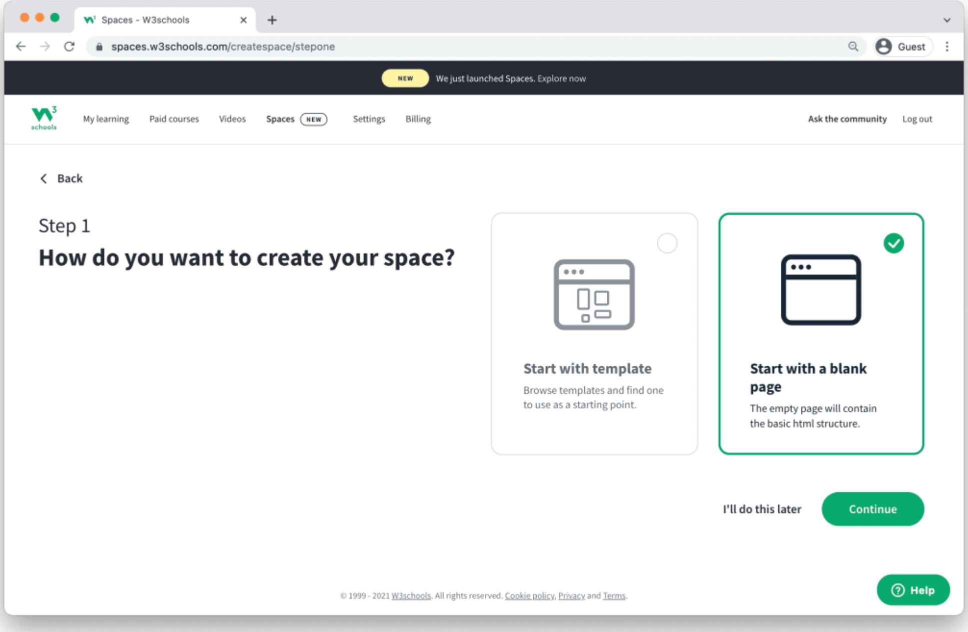
Get more information about how to sign up in our article - How to sign up

Step Two: Start with a template or HTML skeleton

Go to W3Schools Spaces

Select one of the options and click the "**Continue**" button.

Do not worry too much about this decision. You can reset your Space and start over again whenever you want.



Spaces - W3schools
spaces.w3schools.com/createspace/stepone
Guest
NEW
We just launched Spaces. Explore now
My learning
Paid courses
Videos
Spaces NEW
Settings
Billing
Ask the community
Log out
Back

Step 1

How do you want to create your space?

○ Start with template

Browse templates and find one to use as a starting point.

✓ Start with a blank page

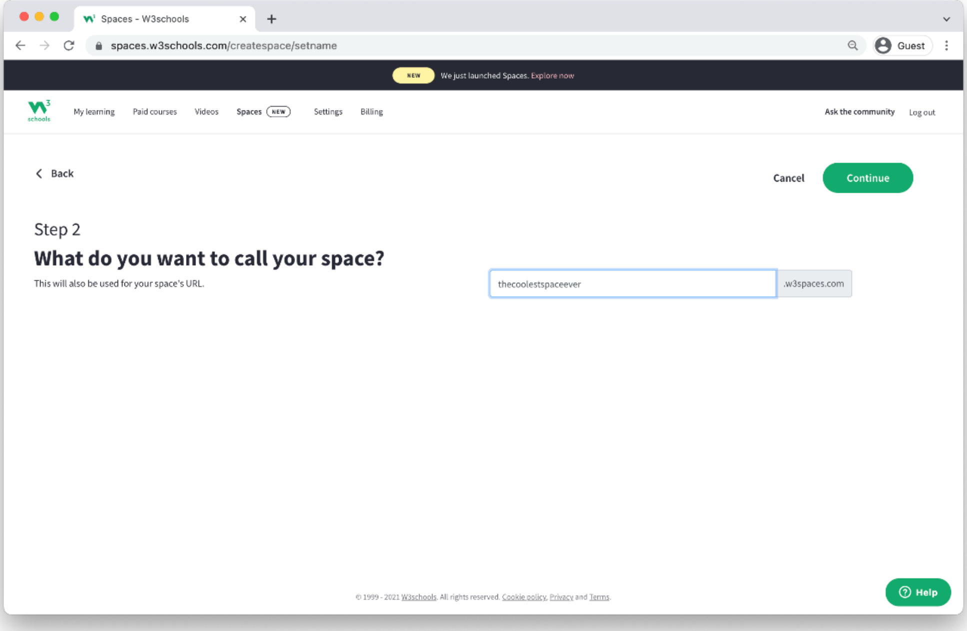
The empty page will contain the basic html structure.

I'll do this later
Continue
Help
© 1999 - 2021 W3schools. All rights reserved. Cookie policy, Privacy and Terms.

Step Three: Give your space a name

Personalize your Space by giving it an amazing name.

You can not use special letters in the name, such as (#, ! or :). The only exception is dash (-)



spaces.w3schools.com/createspace/setname

Step 2

What do you want to call your space?

This will also be used for your space's URL.

thecoolestspaceever .w3spaces.com

© 1999 - 2021 W3schools. All rights reserved. Cookie policy, Privacy and Terms.

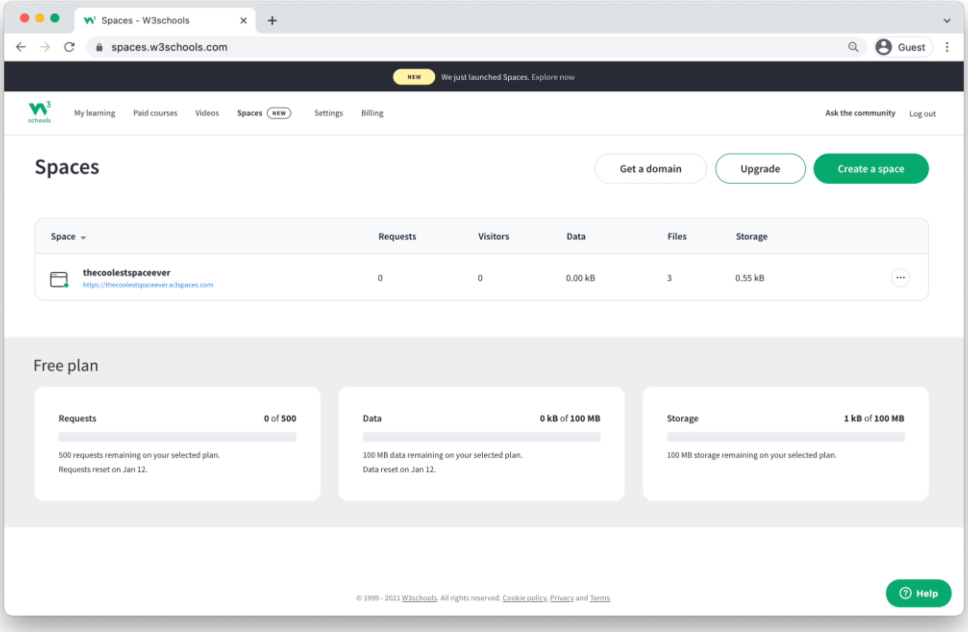
The name will be the link that you share with others to see your site. For example: **yourname**.w3spaces.com

Step Four: Enter your space

Great job! you made it to the dashboard.

In the dashboard you get an overview of your spaces and usage.

Enter your **space** and its **File Overview** by clicking somewhere on the space's row or click the button with the three dots to the right inside the row.



Spaces

Space	Requests	Visitors	Data	Files	Storage
thecoolestspaceever https://thecoolestspaceever.w3spaces.com	0	0	0.00 kB	3	0.55 kB

Free plan

Requests 0 of 500

500 requests remaining on your selected plan.
Requests reset on Jan 12.

Data 0 kB of 100 MB

100 MB data remaining on your selected plan.
Data reset on Jan 12.

Storage 1 kB of 100 MB

100 MB storage remaining on your selected plan.

© 1999 - 2021 W3schools. All rights reserved. Cookie policy, Privacy and Terms.

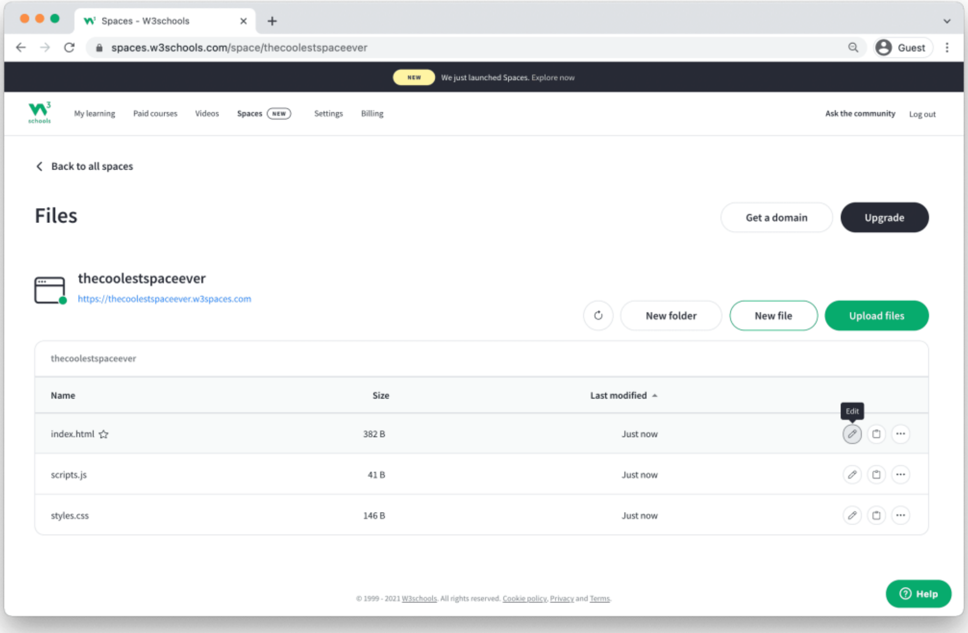
You can only have one space with the free plan. However, you can always upgrade to get more spaces.

Step Five: Edit code or upload files

This is where the magic happens!

Start to **edit code** or **upload files**

- Edit code by clicking on the "**Pen icon**" to the right of the file that you want to edit.
- Create new files by clicking on the "**New file**" button.
- Upload files by clicking on the "**Upload files**" button.



spaces.w3schools.com/space/thecoolestspaceever

We just launched Spaces. Explore now

thecoolestspaceever

https://thecoolestspaceever.w3spaces.com

Name	Size	Last modified
index.html	382 B	Just now
scripts.js	41 B	Just now
styles.css	146 B	Just now

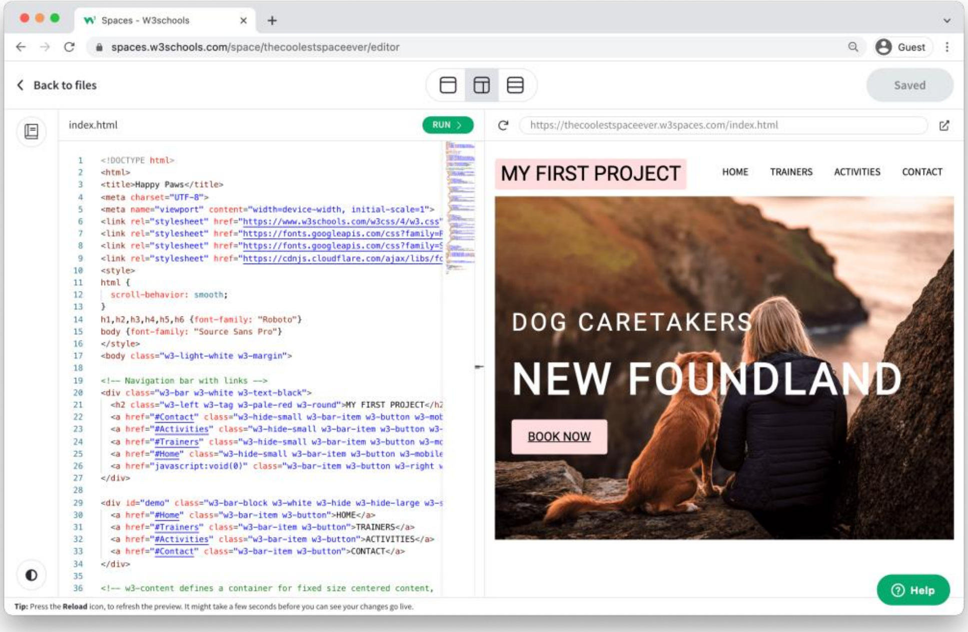
© 1999 - 2021 W3schools. All rights reserved. Cookie policy, Privacy and Terms.

Step Six: Publish your site and share it with someone

This is the start of building your online presence.

Learn, test, build, and go live with your space.

Create your website and share it with others.



```
<!DOCTYPE html>
<html>
<title>Happy Paws</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css"
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=F
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=S
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fc
<style>
html {
  scroll-behavior: smooth;
}
h1,h2,h3,h4,h5,h6 {font-family: "Roboto"}
body {font-family: "Source Sans Pro"}
</style>
<body class="w3-light-white w3-margin">

<!-- Navigation bar with links -->
<div class="w3-bar w3-white w3-text-black">
  <h2 class="w3-left w3-tag w3-pale-red w3-round">MY FIRST PROJECT</h2
  <a href="#Contact" class="w3-hide-small w3-bar-item w3-button w3-mot
  <a href="#Activities" class="w3-hide-small w3-bar-item w3-button w3-
  <a href="#Trainers" class="w3-hide-small w3-bar-item w3-button w3-mc
  <a href="#Home" class="w3-hide-small w3-bar-item w3-button w3-mobile
  <a href="javascript:void(0)" class="w3-bar-item w3-button w3-right w
</div>

<div id="demo" class="w3-bar-block w3-white w3-hide w3-hide-large w3-s
  <a href="#Home" class="w3-bar-item w3-button">HOME</a>
  <a href="#Trainers" class="w3-bar-item w3-button">TRAINERS</a>
  <a href="#Activities" class="w3-bar-item w3-button">ACTIVITIES</a>
  <a href="#Contact" class="w3-bar-item w3-button">CONTACT</a>
</div>

<!-- w3-content defines a container for fixed size centered content,
```

Tip: Press the **Reload** icon, to refresh the preview. It might take a few seconds before you can see your changes go live.

Note: Your space name with the .w3spaces.com extension is your shareable link. Read more about how to share your space in this article [How can I share my space?]

How TO - Make a Static Website

A static website has fixed content

It does not require programming languages to build one.

It is the easiest form of website to create.

Static websites are build of HTML, CSS, and JavaScript.

Get started for free »

Why create a static website?

Static websites are quick and easy to create.

It is cheap to host.

Static websites are secure.

It is fun and you can create awesome sites with HTML, CSS, and JavaScript.

What do I need to know to build a static website

HTML, CSS and JavaScript are the basic languages to build any website.

1. **Create the structure with HTML.** The first thing you have to learn, is HTML, which is the standard markup language for creating web pages.

   Learn HTML »

2. **Style with CSS.** The next step is to learn CSS, to set the layout of your web page with beautiful colors, fonts, and much more.

   Learn CSS »

3. **Make it interactive with JavaScript.** After studying HTML and CSS, you should learn JavaScript to create dynamic and interactive web pages for your users.

   Learn JavaScript »

The best way to learn is to get practical. Start building today!

Create a static website with W3Schools Spaces

Spaces is a personal place where you can build and experiment with code and host your website.

With Spaces you can build static sites with HTML, CSS, and JavaScript.

Everything you need right in the browser.

Learn more »

How do I get started

There are two ways to start building a static website.

Building from **scratch** or using a **template**.

Building a static webpage from scratch

Read here for how to create a static website from scratch How to Create a Webpage

Building with a template

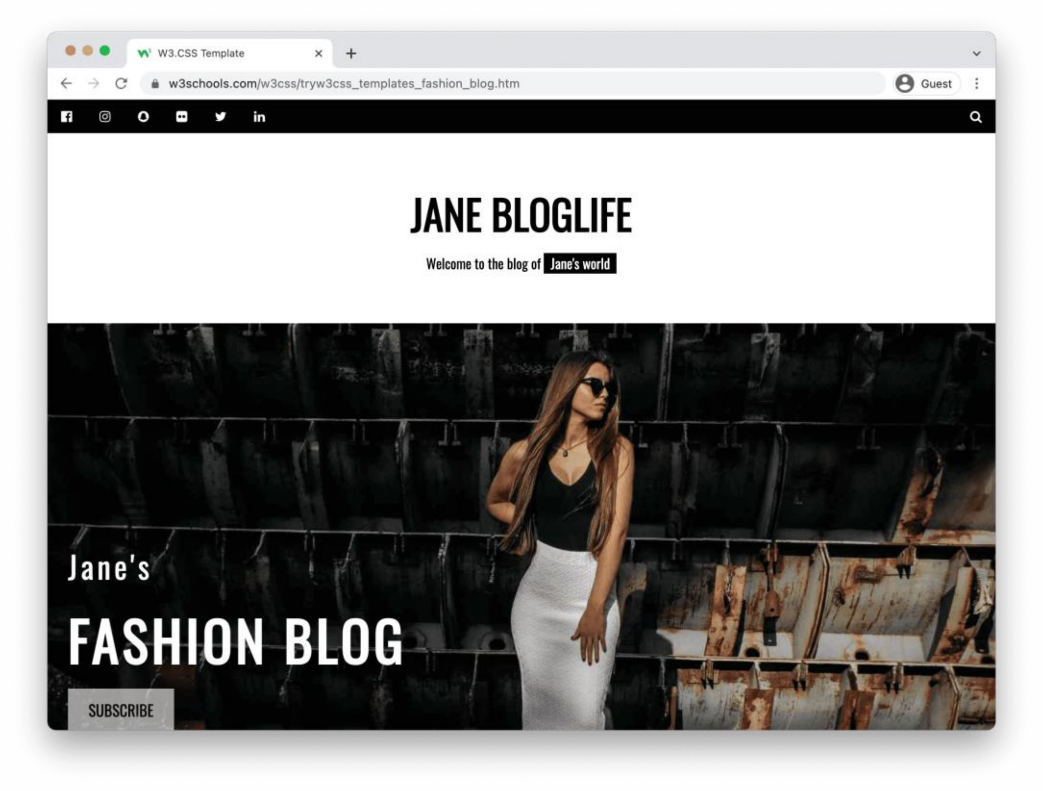
Starting with templates is a good way to get inspired and to learn.

We have ready-made templates that you can use. Here are some examples:

There are many static website templates available in W3Schools Spaces. They can be loaded directly into the service.

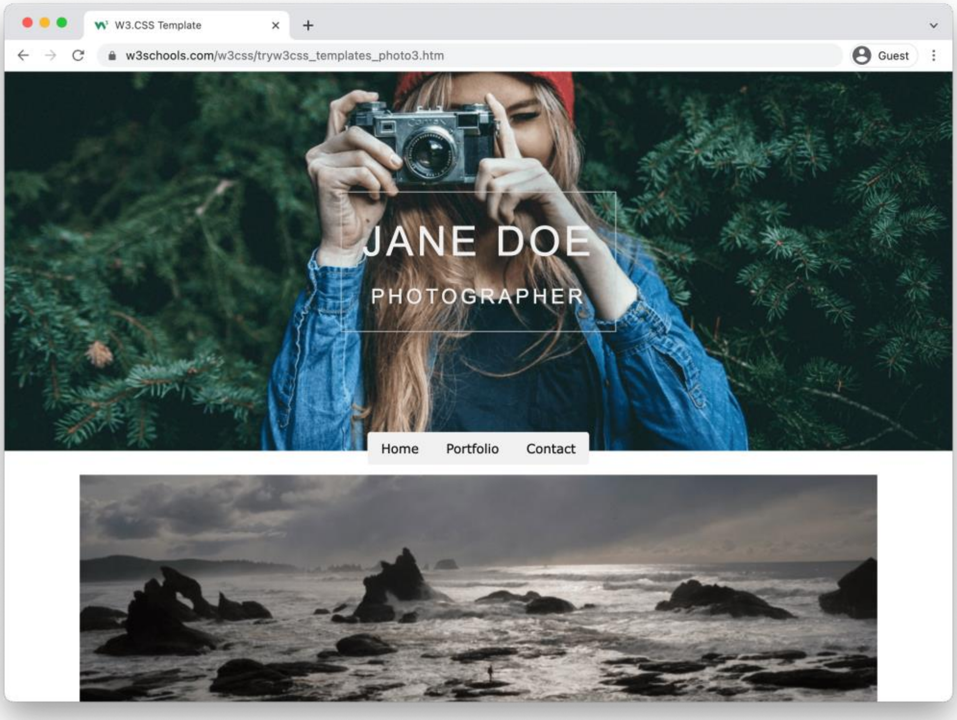
Get started » ** no credit card required*

Fashion Blog Template


W3.CSS Template
w3schools.com/w3css/tryw3css_templates_fashion_blog.htm
Guest
JANE BLOGLIFE
Welcome to the blog of Jane's world
Jane's
FASHION BLOG
SUBSCRIBE

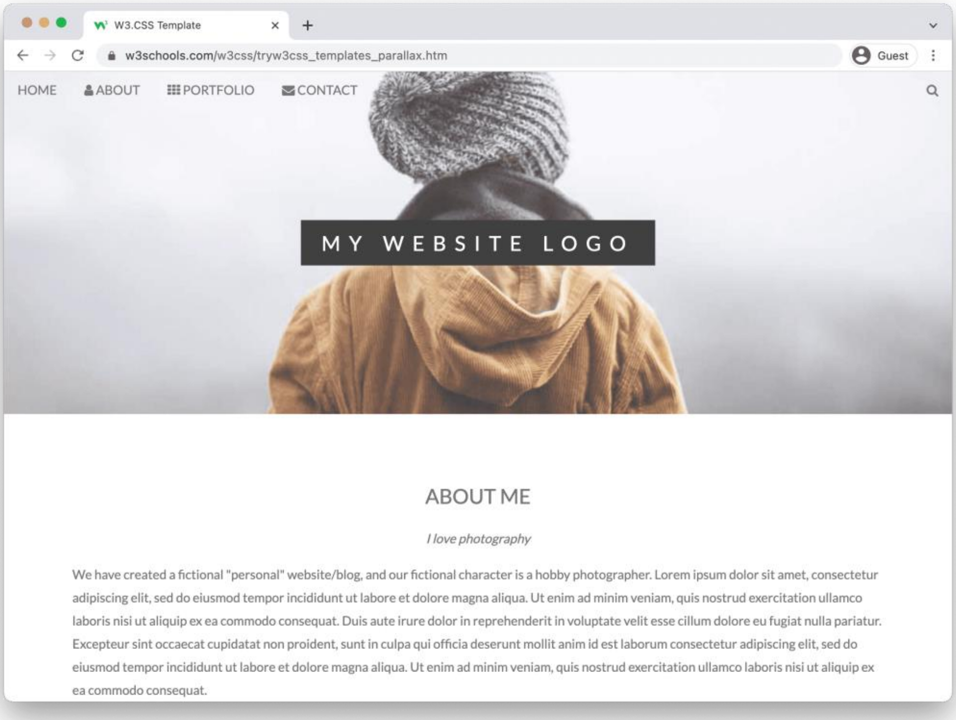
Demo Try it Yourself

Photo Portfolio Template



Demo Try it Yourself

Parallax Template



ABOUT ME

I love photography

We have created a fictional "personal" website/blog, and our fictional character is a hobby photographer. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Demo

Try it Yourself

HTML Introduction

HTML is the standard markup language for creating Web pages.

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

A Simple HTML Document

Example

```html
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Try it Yourself »

Example Explained

- The `<!DOCTYPE html>` declaration defines that this document is an HTML5 document
- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about the HTML page
- The `<title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

- The <h1> element defines a large heading
- The <p> element defines a paragraph

What is an HTML Element?

An HTML element is defined by a start tag, some content, and an end tag:

```
<tagname> Content goes here... </tagname>
```

The HTML **element** is everything from the start tag to the end tag:

```
<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>
```

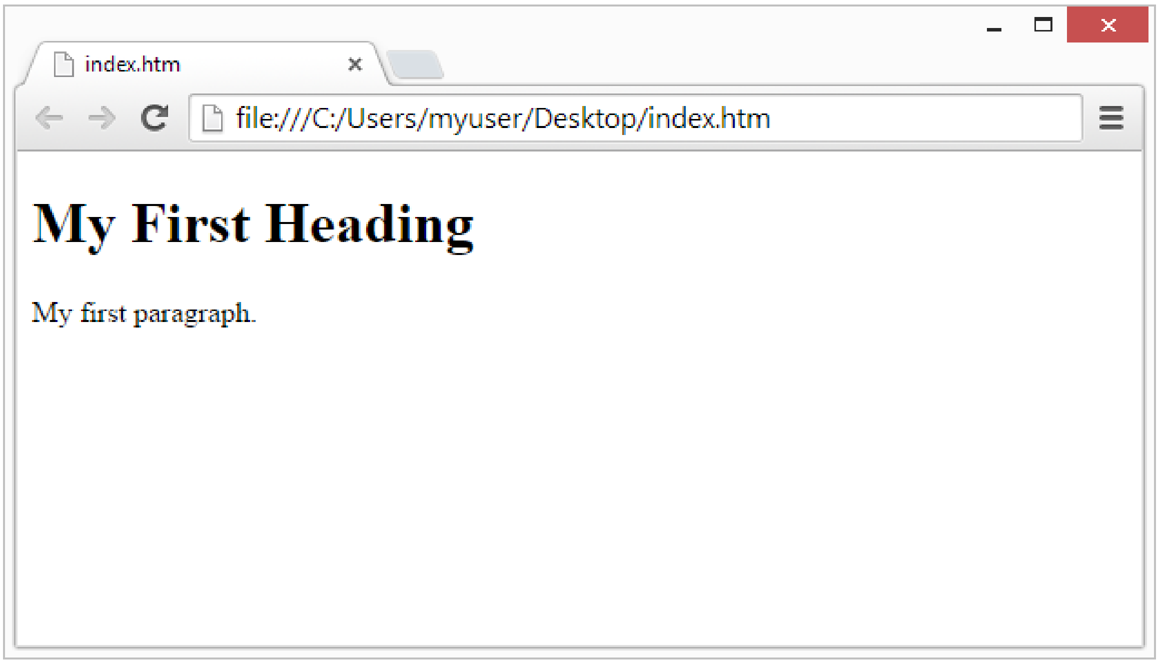
Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>
<br>	*none*	*none*

Note: Some HTML elements have no content (like the <br> element). These elements are called empty elements. Empty elements do not have an end tag!

Web Browsers

The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.

A browser does not display the HTML tags, but uses them to determine how to display the document:



My First Heading

My first paragraph.

HTML Page Structure

Below is a visualization of an HTML page structure:

```
<html>
<head>
<title>Page title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body>
</html>
```

Note: The content inside the <body> section will be displayed in a browser. The content inside the <title> element will be shown in the browser's title bar or in the page's tab.

HTML History

Since the early days of the World Wide Web, there have been many versions of HTML:

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	WHATWG HTML5 Living Standard
2014	W3C Recommendation: HTML5

2016	W3C Candidate Recommendation: HTML 5.1
2017	W3C Recommendation: HTML5.1 2nd Edition
2017	W3C Recommendation: HTML5.2

This tutorial follows the latest HTML5 standard.

HTML Editors

A simple text editor is all you need to learn HTML.

Learn HTML Using Notepad or TextEdit

Web pages can be created and modified by using professional HTML editors.

However, for learning HTML we recommend a simple text editor like Notepad (PC) or TextEdit (Mac).

We believe that using a simple text editor is a good way to learn HTML.

Follow the steps below to create your first web page with Notepad or TextEdit.

Step 1: Open Notepad (PC)

Windows 8 or later:

Open the **Start Screen** (the window symbol at the bottom left on your screen). Type **Notepad**.

Windows 7 or earlier:

Open **Start** > **Programs > Accessories > Notepad**

Step 1: Open TextEdit (Mac)

Open **Finder > Applications > TextEdit**

Also change some preferences to get the application to save files correctly. In **Preferences > Format >** choose **"Plain Text"**

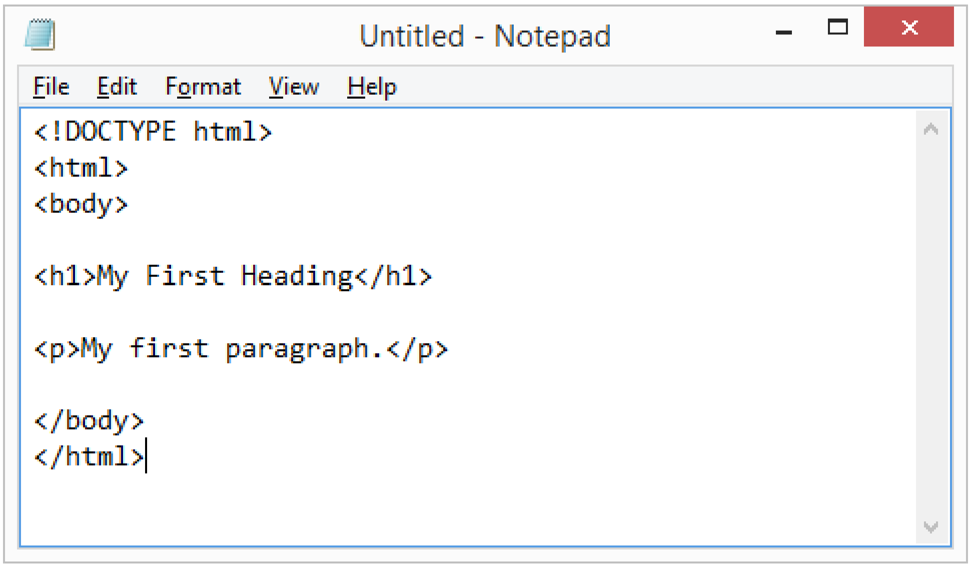
Then under "Open and Save", check the box that says "Display HTML files as HTML code instead of formatted text".

Then open a new document to place the code.

Step 2: Write Some HTML

Write or copy the following HTML code into Notepad:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

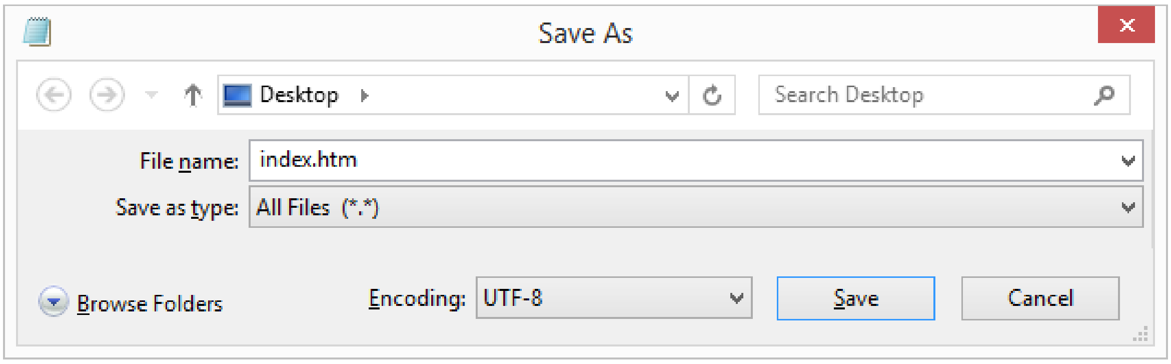


```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

Step 3: Save the HTML Page

Save the file on your computer. Select **File > Save as** in the Notepad menu.

Name the file **"index.htm"** and set the encoding to **UTF-8** (which is the preferred encoding for HTML files).



File name: index.htm

Save as type: All Files (*.*)

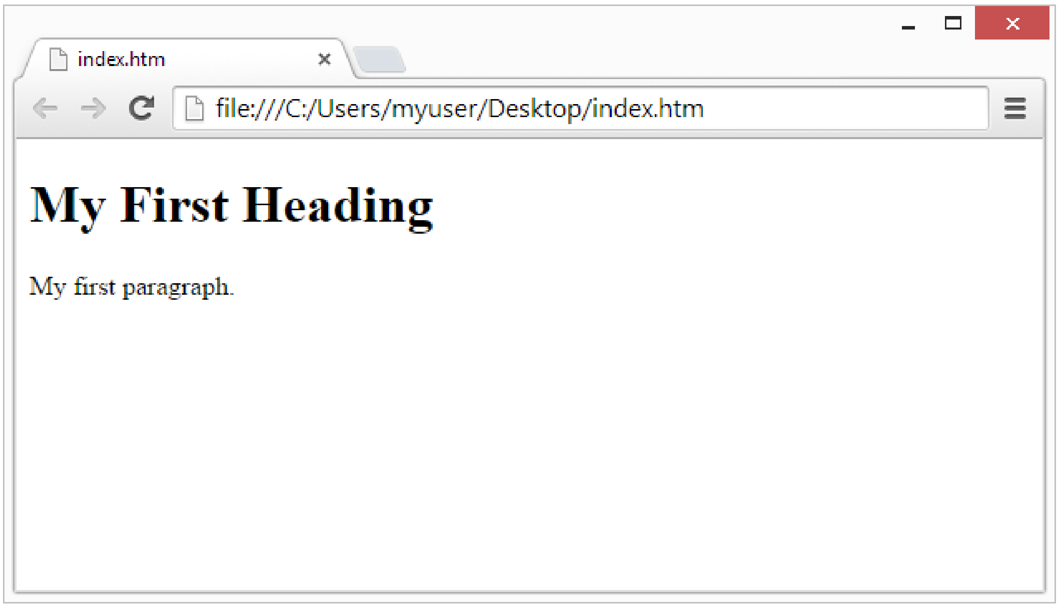
Encoding: UTF-8

Tip: You can use either .htm or .html as file extension. There is no difference; it is up to you.

Step 4: View the HTML Page in Your Browser

Open the saved HTML file in your favorite browser (double click on the file, or right-click - and choose "Open with").

The result will look much like this:



file:///C:/Users/myuser/Desktop/index.htm

My First Heading

My first paragraph.

W3Schools Online Editor - "Try it Yourself"

With our free online editor, you can edit the HTML code and view the result in your browser.

It is the perfect tool when you want to **test** code fast. It also has color coding and the ability to save and share code with others:

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

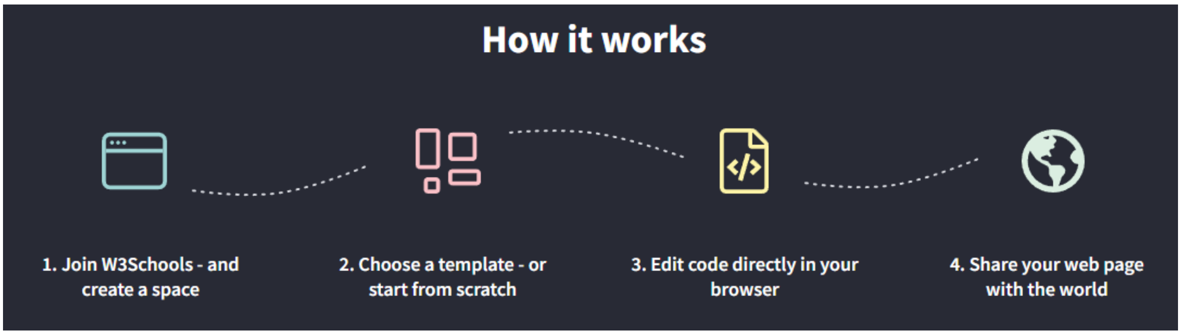
Try it Yourself »

Click on the "Try it Yourself" button to see how it works.

W3Schools Spaces

If you want to create your own website and save your code online, try our free **website builder**, called **W3schools Spaces**:

Get Started Now


How it works
1. Join W3Schools - and create a space
2. Choose a template - or start from scratch
3. Edit code directly in your browser
4. Share your web page with the world

HTML Basic Examples

In this chapter we will show some basic HTML examples.

Don't worry if we use tags you have not learned about yet.

HTML Documents

All HTML documents must start with a document type declaration: `<!DOCTYPE html>`.

The HTML document itself begins with `<html>` and ends with `</html>`.

The visible part of the HTML document is between `<body>` and `</body>`.

Example

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Try it Yourself »

The <!DOCTYPE> Declaration

The `<!DOCTYPE>` declaration represents the document type, and helps browsers to display web pages correctly.

It must only appear once, at the top of the page (before any HTML tags).

The `<!DOCTYPE>` declaration is not case sensitive.

The `<!DOCTYPE>` declaration for HTML5 is:

```
<!DOCTYPE html>
```

HTML Headings

HTML headings are defined with the `<h1>` to `<h6>` tags.

`<h1>` defines the most important heading. `<h6>` defines the least important heading:

Example

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

Try it Yourself »

HTML Paragraphs

HTML paragraphs are defined with the `<p>` tag:

Example

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

Try it Yourself »

HTML Links

HTML links are defined with the `<a>` tag:

Example

```
<a href="https://www.w3schools.com">This is a link</a>
```

Try it Yourself »

The link's destination is specified in the `href` attribute.

Attributes are used to provide additional information about HTML elements.

You will learn more about attributes in a later chapter.

HTML Images

HTML images are defined with the `<img>` tag.

The source file (`src`), alternative text (`alt`), `width`, and `height` are provided as attributes:

Example

```
<img src="w3schools.jpg" alt="W3Schools.com" width="104" height="142">
```

Try it Yourself »

How to View HTML Source

Have you ever seen a Web page and wondered "Hey! How did they do that?"

View HTML Source Code:

Click CTRL + U in an HTML page, or right-click on the page and select "View Page Source". This will open a new tab containing the HTML source code of the page.

Inspect an HTML Element:

Right-click on an element (or a blank area), and choose "Inspect" to see what elements are made up of (you will see both the HTML and the CSS). You can also edit the HTML or CSS on-the-fly in the Elements or Styles panel that opens.

HTML Exercises

Exercise:

HTML elements are surrounded by a specific type of brackets, which one?

```
< p > This is a paragraph. < /p >
```

Start the Exercise

HTML Elements

❮ PreviousNext ❯

An HTML element is defined by a start tag, some content, and an end tag.

HTML Elements

The HTML **element** is everything from the start tag to the end tag:

```
<tagname>Content goes here...</tagname>
```

Examples of some HTML elements:

```
<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>
```

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>
<br>	*none*	*none*

Note: Some HTML elements have no content (like the <br> element). These elements are called empty elements. Empty elements do not have an end tag!

Nested HTML Elements

HTML elements can be nested (this means that elements can contain other elements).

All HTML documents consist of nested HTML elements.

The following example contains four HTML elements (`<html>`, `<body>`, `<h1>` and `<p>`):

Example

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Try it Yourself »

Example Explained

The `<html>` element is the root element and it defines the whole HTML document.

It has a start tag `<html>` and an end tag `</html>`.

Then, inside the `<html>` element there is a `<body>` element:

```
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
```

The `<body>` element defines the document's body.

It has a start tag `<body>` and an end tag `</body>`.

Then, inside the `<body>` element there are two other elements: `<h1>` and `<p>`:

```
<h1>My First Heading</h1>
<p>My first paragraph.</p>
```

The `<h1>` element defines a heading.

It has a start tag `<h1>` and an end tag `</h1>`:

```
<h1>My First Heading</h1>
```

The `<p>` element defines a paragraph.

It has a start tag `<p>` and an end tag `</p>`:

```
<p>My first paragraph.</p>
```

Never Skip the End Tag

Some HTML elements will display correctly, even if you forget the end tag:

Example

```
<html>
<body>

<p>This is a paragraph
<p>This is a paragraph

</body>
</html>
```

Try it Yourself »

However, never rely on this! Unexpected results and errors may occur if you forget the end tag!

Empty HTML Elements

HTML elements with no content are called empty elements.

The `<br>` tag defines a line break, and is an empty element without a closing tag:

Example

```
<p>This is a <br> paragraph with a line break.</p>
```

Try it Yourself »

HTML is Not Case Sensitive

HTML tags are not case sensitive: `<P>` means the same as `<p>`.

The HTML standard does not require lowercase tags, but W3C **recommends** lowercase in HTML, and **demands** lowercase for stricter document types like XHTML.

At W3Schools we always use lowercase tag names.

HTML Exercises

Exercise:

Insert the correct end tag for the HTML heading.

```
<h1>This is a heading</h1>
```

Start the Exercise

HTML Attributes

HTML attributes provide additional information about HTML elements.

HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about elements
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

The href Attribute

The `<a>` tag defines a hyperlink. The `href` attribute specifies the URL of the page the link goes to:

Example

```
<a href="https://www.w3schools.com">Visit W3Schools</a>
```

Try it Yourself »

You will learn more about links in our HTML Links chapter.

The src Attribute

The `<img>` tag is used to embed an image in an HTML page. The `src` attribute specifies the path to the image to be displayed:

Example

```
<img src="img_girl.jpg">
```

Try it Yourself »

There are two ways to specify the URL in the `src` attribute:

1. Absolute URL - Links to an external image that is hosted on another website. Example: src="https://www.w3schools.com/images/img_girl.jpg".

Notes: External images might be under copyright. If you do not get permission to use it, you may be in violation of copyright laws. In addition, you cannot control external images; it can suddenly be removed or changed.

2. Relative URL - Links to an image that is hosted within the website. Here, the URL does not include the domain name. If the URL begins without a slash, it will be relative to the current page. Example: src="img_girl.jpg". If the URL begins with a slash, it will be relative to the domain. Example: src="/images/img_girl.jpg".

Tip: It is almost always best to use relative URLs. They will not break if you change domain.

The width and height Attributes

The `<img>` tag should also contain the `width` and `height` attributes, which specify the width and height of the image (in pixels):

Example

```
<img src="img_girl.jpg" width="500" height="600">
```

Try it Yourself »

The alt Attribute

The required `alt` attribute for the `<img>` tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to a slow connection, or an error in the `src` attribute, or if the user uses a screen reader.

Example

```
<img src="img_girl.jpg" alt="Girl with a jacket">
```

Try it Yourself »

Example

See what happens if we try to display an image that does not exist:

```
<img src="img_typo.jpg" alt="Girl with a jacket">
```

Try it Yourself »

You will learn more about images in our HTML Images chapter.

The style Attribute

The `style` attribute is used to add styles to an element, such as color, font, size, and more.

Example

```
<p style="color:red;">This is a red paragraph.</p>
```

Try it Yourself »

You will learn more about styles in our HTML Styles chapter.

The lang Attribute

You should always include the `lang` attribute inside the `<html>` tag, to declare the language of the Web page. This is meant to assist search engines and browsers.

The following example specifies English as the language:

```
<!DOCTYPE html>
<html lang="en">
<body>
...
</body>
</html>
```

Country codes can also be added to the language code in the `lang` attribute. So, the first two characters define the language of the HTML page, and the last two characters define the country.

The following example specifies English as the language and United States as the country:

```
<!DOCTYPE html>
<html lang="en-US">
<body>
...
</body>
</html>
```

You can see all the language codes in our HTML Language Code Reference.

The title Attribute

The `title` attribute defines some extra information about an element.

The value of the title attribute will be displayed as a tooltip when you mouse over the element:

Example

```
<p title="I'm a tooltip">This is a paragraph.</p>
```

Try it Yourself »

We Suggest: Always Use Lowercase Attributes

The HTML standard does not require lowercase attribute names.

The title attribute (and all other attributes) can be written with uppercase or lowercase like **title** or **TITLE**.

However, W3C **recommends** lowercase attributes in HTML,
and **demands** lowercase attributes for stricter document types like XHTML.

At W3Schools we always use lowercase attribute names.

We Suggest: Always Quote Attribute Values

The HTML standard does not require quotes around attribute values.

However, W3C **recommends** quotes in HTML, and **demands** quotes for stricter document types like XHTML.

Good:

```
<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

Bad:

```
<a href=https://www.w3schools.com/html/>Visit our HTML tutorial</a>
```

Sometimes you have to use quotes. This example will not display the title attribute correctly, because it contains a space:

Example

```
<p title=About W3Schools>
```

Try it Yourself »

At W3Schools we always use quotes around attribute values.

Single or Double Quotes?

Double quotes around attribute values are the most common in HTML, but single quotes can also be used.

In some situations, when the attribute value itself contains double quotes, it is necessary to use single quotes:

```
<p title='John "ShotGun" Nelson'>
```

Or vice versa:

```
<p title="John 'ShotGun' Nelson">
```

Try it Yourself »

Chapter Summary

- All HTML elements can have **attributes**
- The `href` attribute of `<a>` specifies the URL of the page the link goes to
- The `src` attribute of `<img>` specifies the path to the image to be displayed
- The `width` and `height` attributes of `<img>` provide size information for images
- The `alt` attribute of `<img>` provides an alternate text for an image
- The `style` attribute is used to add styles to an element, such as color, font, size, and more
- The `lang` attribute of the `<html>` tag declares the language of the Web page
- The `title` attribute defines some extra information about an element

HTML Exercises

Exercise:

Add a "tooltip" to the paragraph below with the text "About W3Schools".

```
<p          ="About W3Schools">W3Schools is a web developer's
site.</p>
```

Start the Exercise

HTML Attribute Reference

A complete list of all attributes for each HTML element, is listed in our: HTML Attribute Reference.

HTML Headings

HTML headings are titles or subtitles that you want to display on a webpage.

Example

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Try it Yourself »

HTML Headings

HTML headings are defined with the `<h1>` to `<h6>` tags.

`<h1>` defines the most important heading. `<h6>` defines the least important heading.

Example

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

Try it Yourself »

Note: Browsers automatically add some white space (a margin) before and after a heading.

Headings Are Important

Search engines use the headings to index the structure and content of your web pages.

Users often skim a page by its headings. It is important to use headings to show the document structure.

`<h1>` headings should be used for main headings, followed by `<h2>` headings, then the less important `<h3>`, and so on.

Note: Use HTML headings for headings only. Don't use headings to make text **BIG** or **bold**.

Bigger Headings

Each HTML heading has a default size. However, you can specify the size for any heading with the `style` attribute, using the CSS `font-size` property:

Example

```
<h1 style="font-size:60px;">Heading 1</h1>
```

Try it Yourself »

HTML Exercises

Exercise:

Use the correct HTML tag to add a heading with the text "London".

□

```
<p>London is the capital city of England. It is the most
populous city in the United Kingdom, with a metropolitan
area of over 13 million inhabitants.</p>
```

Start the Exercise

HTML Tag Reference

W3Schools' tag reference contains additional information about these tags and their attributes.

Tag	Description
<html>	Defines the root of an HTML document

<body>	Defines the document's body
<h1> to <h6>	Defines HTML headings

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Paragraphs

A paragraph always starts on a new line, and is usually a block of text.

HTML Paragraphs

The HTML `<p>` element defines a paragraph.

A paragraph always starts on a new line, and browsers automatically add some white space (a margin) before and after a paragraph.

Example

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

Try it Yourself »

HTML Display

You cannot be sure how HTML will be displayed.

Large or small screens, and resized windows will create different results.

With HTML, you cannot change the display by adding extra spaces or extra lines in your HTML code.

The browser will automatically remove any extra spaces and lines when the page is displayed:

Example

```
<p>
This paragraph
contains a lot of lines
in the source code,
but the browser
ignores it.
</p>

<p>
This paragraph
contains         a lot of spaces
in the source         code,
but the        browser
ignores it.
</p>
```

Try it Yourself »

HTML Horizontal Rules

The `<hr>` tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

The `<hr>` element is used to separate content (or define a change) in an HTML page:

Example

```
<h1>This is heading 1</h1>
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
```

Try it Yourself »

The `<hr>` tag is an empty tag, which means that it has no end tag.

HTML Line Breaks

The HTML `<br>` element defines a line break.

Use `<br>` if you want a line break (a new line) without starting a new paragraph:

Example

```
<p>This is<br>a paragraph<br>with line breaks.</p>
```

Try it Yourself »

The <br> tag is an empty tag, which means that it has no end tag.

The Poem Problem

This poem will display on a single line:

Example

```
<p>
  My Bonnie lies over the ocean.

  My Bonnie lies over the sea.

  My Bonnie lies over the ocean.

  Oh, bring back my Bonnie to me.
</p>
```

Try it Yourself »

Solution - The HTML <pre> Element

The HTML <pre> element defines preformatted text.

The text inside a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks:

Example

```
<pre>
  My Bonnie lies over the ocean.

  My Bonnie lies over the sea.

  My Bonnie lies over the ocean.

  Oh, bring back my Bonnie to me.
</pre>
```

Try it Yourself »

HTML Exercises

Exercise:

Use the correct HTML tag to add a paragraph with the text "Hello World!".

```
<html>
<body>

</body>
</html>
```

Start the Exercise

HTML Tag Reference

W3Schools' tag reference contains additional information about HTML elements and their attributes.

Tag	Description
<p>	Defines a paragraph
<hr>	Defines a thematic change in the content
<br>	Inserts a single line break
<pre>	Defines pre-formatted text

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Styles

The HTML `style` attribute is used to add styles to an element, such as color, font, size, and more.

Example

I am Red

I am Blue

I am Big

Try it Yourself »

The HTML Style Attribute

Setting the style of an HTML element, can be done with the `style` attribute.

The HTML `style` attribute has the following syntax:

```
<tagname style="property:value;">
```

The ***property*** is a CSS property. The ***value*** is a CSS value.

You will learn more about CSS later in this tutorial.

Background Color

The CSS `background-color` property defines the background color for an HTML element.

Example

Set the background color for a page to powderblue:

```
<body style="background-color:powderblue;">

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
```

Try it Yourself »

Example

Set background color for two different elements:

```
<body>

<h1 style="background-color:powderblue;">This is a heading</h1>
<p style="background-color:tomato;">This is a paragraph.</p>

</body>
```

Try it Yourself »

Text Color

The CSS `color` property defines the text color for an HTML element:

Example

```
<h1 style="color:blue;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```

Try it Yourself »

Fonts

The CSS `font-family` property defines the font to be used for an HTML element:

Example

```
<h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:courier;">This is a paragraph.</p>
```

Try it Yourself »

Text Size

The CSS `font-size` property defines the text size for an HTML element:

Example

```
<h1 style="font-size:300%;">This is a heading</h1>
<p style="font-size:160%;">This is a paragraph.</p>
```

Try it Yourself »

Text Alignment

The CSS `text-align` property defines the horizontal text alignment for an HTML element:

Example

```
<h1 style="text-align:center;">Centered Heading</h1>
<p style="text-align:center;">Centered paragraph.</p>
```

Try it Yourself »

Chapter Summary

- Use the `style` attribute for styling HTML elements
- Use `background-color` for background color
- Use `color` for text colors
- Use `font-family` for text fonts
- Use `font-size` for text sizes
- Use `text-align` for text alignment

HTML Exercises

Exercise:

Use the correct HTML attribute, and CSS, to set the colour of the paragraph to "blue".

```
<p □="□;">This is a paragraph.</p>
```

Start the Exercise

HTML Text Formatting

HTML contains several elements for defining text with a special meaning.

Example

This text is bold

This text is italic

This is $_{\text{subscript}}$ and $^{\text{superscript}}$

Try it Yourself »

HTML Formatting Elements

Formatting elements were designed to display special types of text:

- `<b>` - Bold text
- `<strong>` - Important text
- `<i>` - Italic text
- `<em>` - Emphasized text
- `<mark>` - Marked text
- `<small>` - Smaller text
- `<del>` - Deleted text
- `<ins>` - Inserted text
- `<sub>` - Subscript text

- `<sup>` - Superscript text

HTML <b> and <strong> Elements

The HTML `<b>` element defines bold text, without any extra importance.

Example

```
<b>This text is bold</b>
```

Try it Yourself »

The HTML `<strong>` element defines text with strong importance. The content inside is typically displayed in bold.

Example

```
<strong>This text is important!</strong>
```

Try it Yourself »

HTML <i> and <em> Elements

The HTML `<i>` element defines a part of text in an alternate voice or mood. The content inside is typically displayed in italic.

Tip: The `<i>` tag is often used to indicate a technical term, a phrase from another language, a thought, a ship name, etc.

Example

```
<i>This text is italic</i>
```

Try it Yourself »

The HTML `<em>` element defines emphasized text. The content inside is typically displayed in italic.

Tip: A screen reader will pronounce the words in `<em>` with an emphasis, using verbal stress.

Example

```
<em>This text is emphasized</em>
```

Try it Yourself »

HTML <small> Element

The HTML `<small>` element defines smaller text:

Example

```
<small>This is some smaller text.</small>
```

Try it Yourself »

HTML <mark> Element

The HTML `<mark>` element defines text that should be marked or highlighted:

Example

```
<p>Do not forget to buy <mark>milk</mark> today.</p>
```

Try it Yourself »

HTML <del> Element

The HTML `<del>` element defines text that has been deleted from a document. Browsers will usually strike a line through deleted text:

Example

```
<p>My favorite color is <del>blue</del> red.</p>
```

Try it Yourself »

HTML <ins> Element

The HTML `<ins>` element defines a text that has been inserted into a document. Browsers will usually underline inserted text:

Example

```
<p>My favorite color is <del>blue</del> <ins>red</ins>.</p>
```

Try it Yourself »

HTML <sub> Element

The HTML `<sub>` element defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H_2O:

Example

```
<p>This is <sub>subscripted</sub> text.</p>
```

Try it Yourself »

HTML <sup> Element

The HTML `<sup>` element defines superscript text. Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font. Superscript text can be used for footnotes, like WWW[1]:

Example

```
<p>This is <sup>superscripted</sup> text.</p>
```

Try it Yourself »

HTML Exercises

Exercise:

Add extra importance to the word "degradation" in the paragraph below.

```
<p>
WWF's mission is to stop the □degradation□ of our
planet's natural environment.
</p>
```

Start the Exercise

HTML Text Formatting Elements

Tag	Description
<b>	Defines bold text
<em>	Defines emphasized text

<i>	Defines a part of text in an alternate voice or mood
<small>	Defines smaller text
<strong>	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<ins>	Defines inserted text
<del>	Defines deleted text
<mark>	Defines marked/highlighted text

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Quotation and Citation Elements

In this chapter we will go through the `<blockquote>`,`<q>`, `<abbr>`, `<address>`, `<cite>`, and `<bdo>` HTML elements.

Example

Here is a quote from WWF's website:

For 60 years, WWF has worked to help people and nature thrive. As the world's leading conservation organization, WWF works in nearly 100 countries. At every level, we collaborate with people around the world to develop and deliver innovative solutions that protect communities, wildlife, and the places in which they live.

Try it Yourself »

HTML <blockquote> for Quotations

The HTML `<blockquote>` element defines a section that is quoted from another source.

Browsers usually indent `<blockquote>` elements.

Example

```
<p>Here is a quote from WWF's website:</p>
<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 60 years, WWF has worked to help people and nature thrive. As the
world's leading conservation organization, WWF works in nearly 100
countries. At every level, we collaborate with people around the world
to develop and deliver innovative solutions that protect communities,
wildlife, and the places in which they live.
</blockquote>
```

Try it Yourself »

HTML <q> for Short Quotations

The HTML `<q>` tag defines a short quotation.

Browsers normally insert quotation marks around the quotation.

Example

```
<p>WWF's goal is to: <q>Build a future where people live in harmony
with nature.</q></p>
```

Try it Yourself »

HTML <abbr> for Abbreviations

The HTML `<abbr>` tag defines an abbreviation or an acronym, like "HTML", "CSS", "Mr.", "Dr.", "ASAP", "ATM".

Marking abbreviations can give useful information to browsers, translation systems and search-engines.

Tip: Use the global title attribute to show the description for the abbreviation/acronym when you mouse over the element.

Example

```
<p>The <abbr title="World Health Organization">WHO</abbr> was founded
in 1948.</p>
```

Try it Yourself »

HTML <address> for Contact Information

The HTML `<address>` tag defines the contact information for the author/owner of a document or an article.

The contact information can be an email address, URL, physical address, phone number, social media handle, etc.

The text in the `<address>` element usually renders in *italic,* and browsers will always add a line break before and after the `<address>` element.

Example

```
<address>
Written by John Doe.<br>
Visit us at:<br>
Example.com<br>
Box 564, Disneyland<br>
USA
</address>
```

Try it Yourself »

HTML <cite> for Work Title

The HTML `<cite>` tag defines the title of a creative work (e.g. a book, a poem, a song, a movie, a painting, a sculpture, etc.).

Note: A person's name is not the title of a work.

The text in the `<cite>` element usually renders in *italic*.

Example

```
<p><cite>The Scream</cite> by Edvard Munch. Painted in 1893.</p>
```

Try it Yourself »

HTML <bdo> for Bi-Directional Override

BDO stands for Bi-Directional Override.

The HTML `<bdo>` tag is used to override the current text direction:

Example

```
<bdo dir="rtl">This text will be written from right to left</bdo>
```

Try it Yourself »

HTML Exercises

Exercise:

Use an HTML element to add quotation marks around the letters "cool".

```
<p>
I am so <q> cool</q> .
</p>
```

Start the Exercise

HTML Quotation and Citation Elements

Tag	Description
<abbr>	Defines an abbreviation or acronym
<address>	Defines contact information for the author/owner of a document

<bdo>	Defines the text direction
<blockquote>	Defines a section that is quoted from another source
<cite>	Defines the title of a work
<q>	Defines a short inline quotation

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Comments

HTML comments are not displayed in the browser, but they can help document your HTML source code.

HTML Comment Tag

You can add comments to your HTML source by using the following syntax:

```
<!-- Write your comments here -->
```

Notice that there is an exclamation point (!) in the start tag, but not in the end tag.

Note: Comments are not displayed by the browser, but they can help document your HTML source code.

Add Comments

With comments you can place notifications and reminders in your HTML code:

Example

```
<!-- This is a comment -->

<p>This is a paragraph.</p>
```

```
<!-- Remember to add more information here -->
```

Try it Yourself »

Hide Content

Comments can be used to hide content.

This can be helpful if you hide content temporarily:

Example

```
<p>This is a paragraph.</p>

<!-- <p>This is another paragraph </p> -->

<p>This is a paragraph too.</p>
```

Try it Yourself »

You can also hide more than one line. Everything between the `<!--` and the `-->` will be hidden from the display.

Example

Hide a section of HTML code:

```
<p>This is a paragraph.</p>
<!--
<p>Look at this cool image:</p>
<img border="0" src="pic_trulli.jpg" alt="Trulli">
-->
<p>This is a paragraph too.</p>
```

Try it Yourself »

Comments are also great for debugging HTML, because you can comment out HTML lines of code, one at a time, to search for errors.

Hide Inline Content

Comments can be used to hide parts in the middle of the HTML code.

Example

Hide a part of a paragraph:

```
<p>This <!-- great text --> is a paragraph.</p>
```

Try it Yourself »

HTML Exercises

Exercise:

Use the HTML comment tag to make a comment out of the "This is a comment" text.

```
<h1>This is a heading</h1>
□ This is a comment □
<p>This is a paragraph.</p>
```

Start the Exercise

HTML Colours

HTML colours are specified with predefined colour names, or with RGB, HEX, HSL, RGBA, or HSLA values.

Colour Names

In HTML, a colour can be specified by using a colour name:

Tomato

Orange

DodgerBlue

MediumSeaGreen

Gray

SlateBlue

Violet

LightGray

Try it Yourself »

HTML supports 140 standard color names.

Background Colour

You can set the background colour for HTML elements:

Hello World

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Example

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

Try it Yourself »

Text Colour

You can set the colour of text:

Hello World

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Example

```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

Try it Yourself »

Border Colour

You can set the colour of borders:

Hello World

Hello World

Hello World

Example

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

Try it Yourself »

Colour Values

In HTML, colours can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values.

The following three <div> elements have their background colour set with RGB, HEX, and HSL values:

rgb(255, 99, 71)
#ff6347
hsl(9, 100%, 64%)

The following two <div> elements have their background colour set with RGBA and HSLA values, which add an Alpha channel to the colour (here we have 50% transparency):

rgba(255, 99, 71, 0.5)

hsla(9, 100%, 64%, 0.5)

Example

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
<h1 style="background-color:#ff6347;">...</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>

<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

Try it Yourself »

HTML Exercises

Exercise:

Insert the correct property to make the text colour violet.

```
<p style="          : violet">This is a paragraph.</p>
```

Start the Exercise

HTML RGB and RGBA Colors

An RGB color value represents RED, GREEN, and BLUE light sources.

An RGBA color value is an extension of RGB with an Alpha channel (opacity).

RGB Color Values

In HTML, a color can be specified as an RGB value, using this formula:

rgb(*red*, *green*, *blue*)

Each parameter (red, green, and blue) defines the intensity of the color with a value between 0 and 255.

This means that there are 256 x 256 x 256 = 16777216 possible colors!

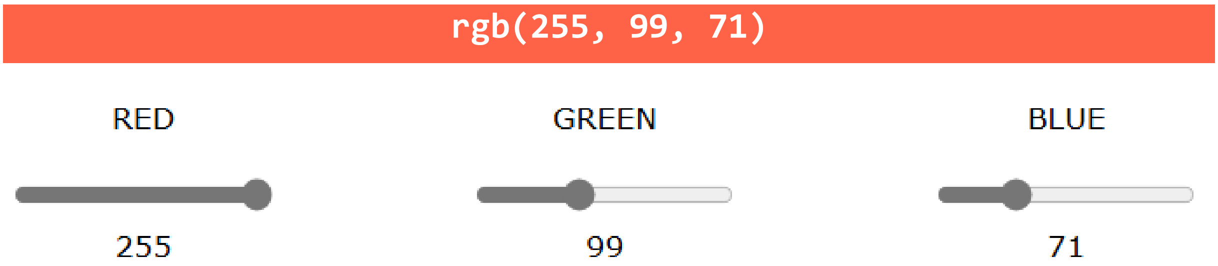
For example, rgb(255, 0, 0) is displayed as red, because red is set to its highest value (255), and the other two (green and blue) are set to 0.

Another example, rgb(0, 255, 0) is displayed as green, because green is set to its highest value (255), and the other two (red and blue) are set to 0.

To display black, set all color parameters to 0, like this: rgb(0, 0, 0).

To display white, set all color parameters to 255, like this: rgb(255, 255, 255).

Experiment by mixing the RGB values below:



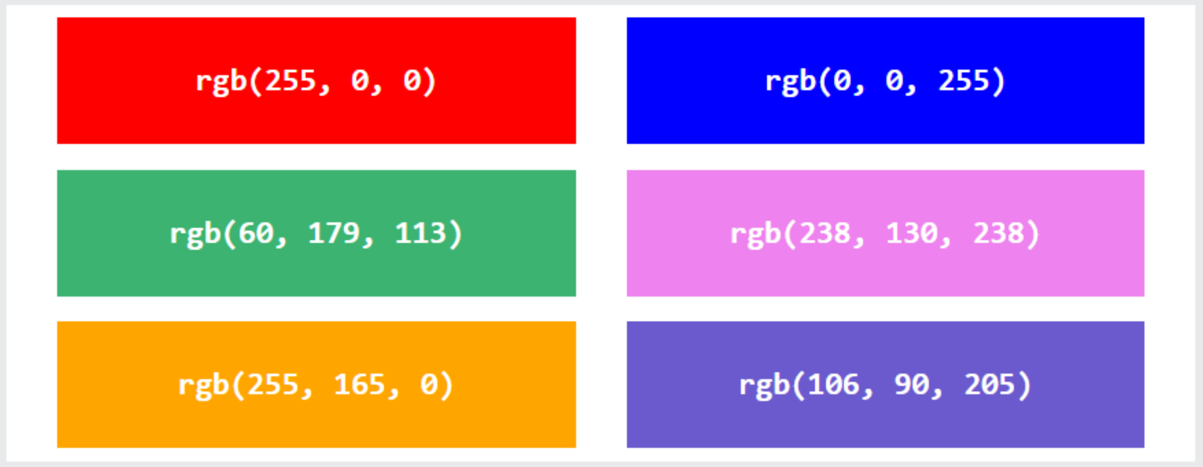
rgb(255, 99, 71)

RED 255

GREEN 99

BLUE 71

Example



rgb(255, 0, 0)

rgb(0, 0, 255)

rgb(60, 179, 113)

rgb(238, 130, 238)

rgb(255, 165, 0)

rgb(106, 90, 205)

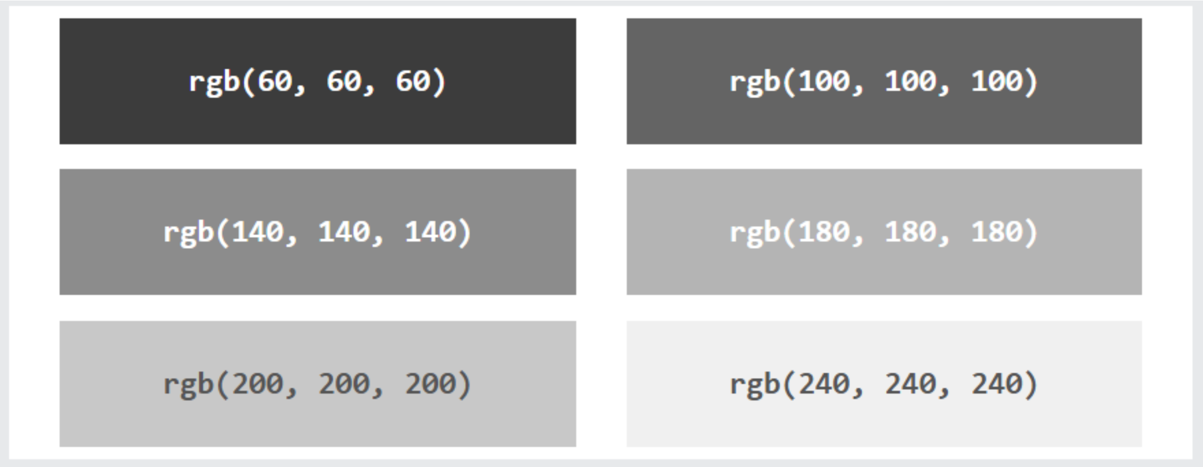
rgb(255, 0, 0)
rgb(0, 0, 255)
rgb(60, 179, 113)
rgb(238, 130, 238)
rgb(255, 165, 0)
rgb(106, 90, 205)

Try it Yourself »

Shades of Gray

Shades of gray are often defined using equal values for all three parameters:

Example



rgb(60, 60, 60)

rgb(100, 100, 100)

rgb(140, 140, 140)

rgb(180, 180, 180)

rgb(200, 200, 200)

rgb(240, 240, 240)

rgb(60, 60, 60)
rgb(100, 100, 100)
rgb(140, 140, 140)
rgb(180, 180, 180)
rgb(200, 200, 200)

```
rgb(240, 240, 240)
```

Try it Yourself »

RGBA Color Values

RGBA color values are an extension of RGB color values with an Alpha channel - which specifies the opacity for a color.

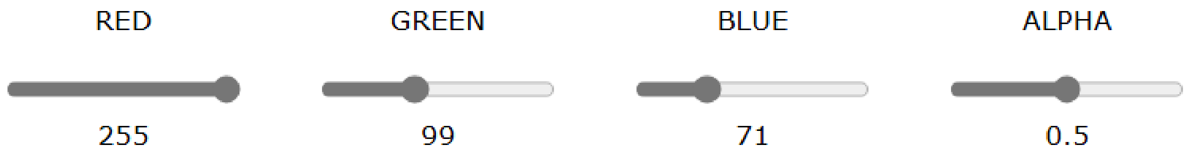
An RGBA color value is specified with:

rgba(*red*, *green*, *blue*, *alpha*)

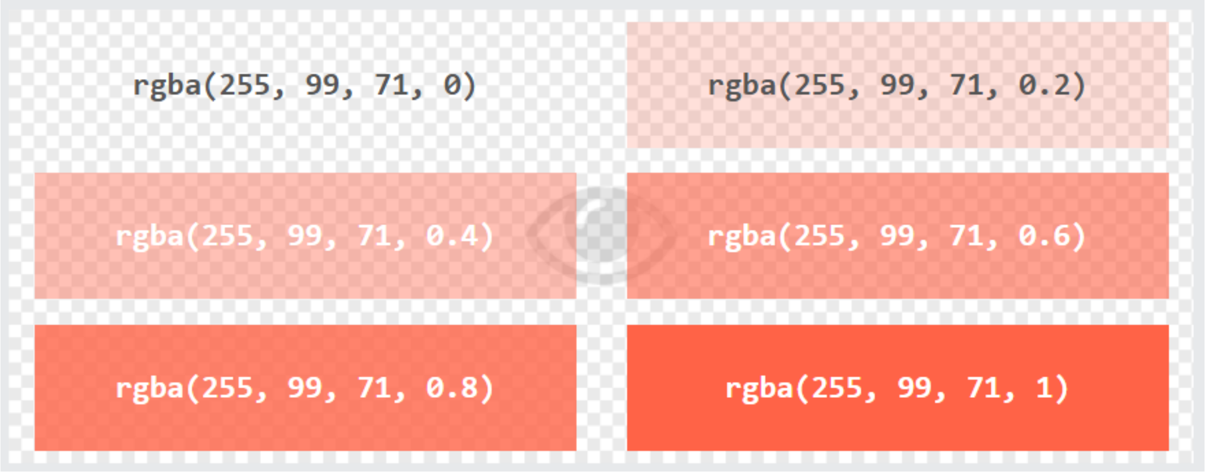
The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

Experiment by mixing the RGBA values below:

rgba(255, 99, 71, 0.5)



Example



rgba(255, 99, 71, 0)
rgba(255, 99, 71, 0.2)
rgba(255, 99, 71, 0.4)
rgba(255, 99, 71, 0.6)
rgba(255, 99, 71, 0.8)
rgba(255, 99, 71, 1)

```
rgba(255, 99, 71, 0)
rgba(255, 99, 71, 0.2)
rgba(255, 99, 71, 0.4)
rgba(255, 99, 71, 0.6)
rgba(255, 99, 71, 0.8)
```

rgba(255, 99, 71, 1)

Try it Yourself »

HTML Exercises

Exercise:

Insert the correct RGB color values to make the background color completely blue.

```
<p style="background-color:rgb(□, □, □)">This
is a paragraph.</p>
```

Start the Exercise

HTML HEX Colors

A hexadecimal color is specified with: #RRGGBB, where the RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color.

HEX Color Values

In HTML, a color can be specified using a hexadecimal value in the form:

#rrggbb

Where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255).

For example, #ff0000 is displayed as red, because red is set to its highest value (ff), and the other two (green and blue) are set to 00.

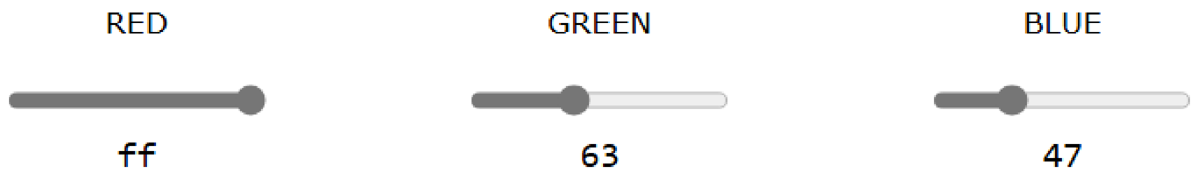
Another example, #00ff00 is displayed as green, because green is set to its highest value (ff), and the other two (red and blue) are set to 00.

To display black, set all color parameters to 00, like this: #000000.

To display white, set all color parameters to ff, like this: #ffffff.

Experiment by mixing the HEX values below:

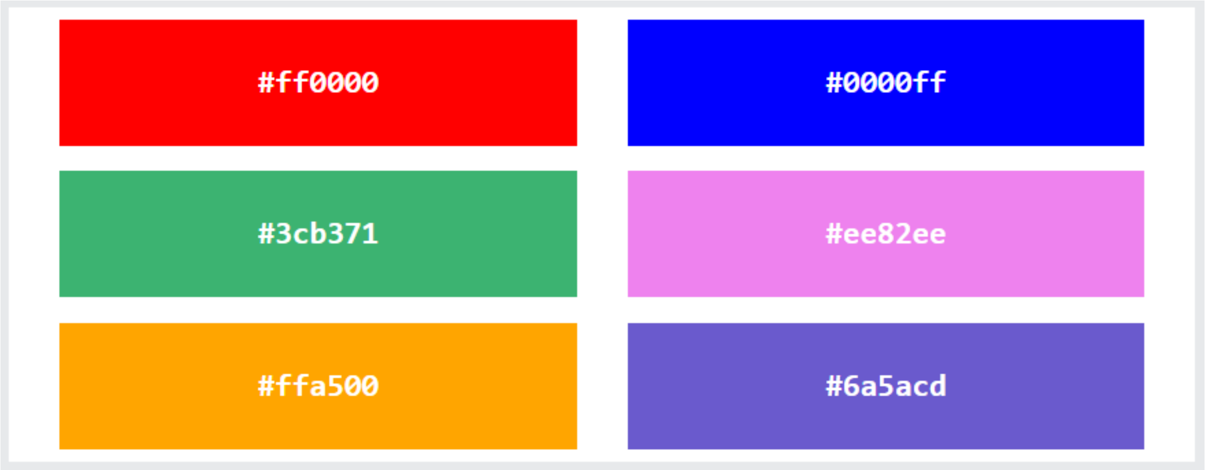
#ff6347


RED ff

GREEN 63

BLUE 47

Example


#ff0000

#0000ff

#3cb371

#ee82ee

#ffa500

#6a5acd

#ff0000

#0000ff

#3cb371

#ee82ee

#ffa500

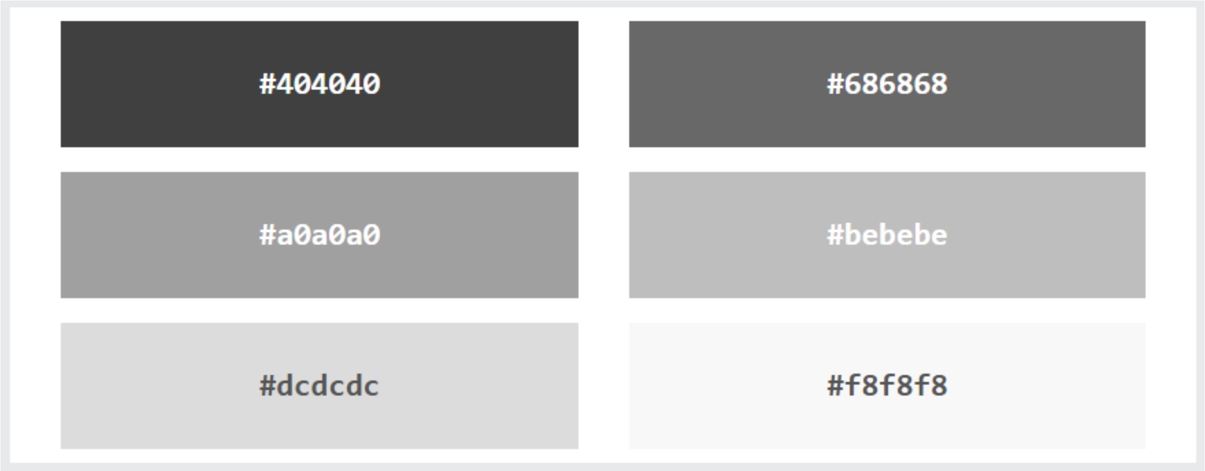
#6a5acd

Try it Yourself »

Shades of Gray

Shades of gray are often defined using equal values for all three parameters:

Example



#404040 #686868

#a0a0a0 #bebebe

#dcdcdc #f8f8f8

#404040

#686868

#a0a0a0

#bebebe

#dcdcdc

#f8f8f8

Try it Yourself »

HTML Exercises

Exercise:

Insert the correct HEX value to make the text color white.

```
<p style="color: #□">This is a paragraph.</p>
```

Start the Exercise

HTML HSL and HSLA Colors

HSL stands for hue, saturation, and lightness.

HSLA color values are an extension of HSL with an Alpha channel (opacity).

HSL Color Values

In HTML, a color can be specified using hue, saturation, and lightness (HSL) in the form:

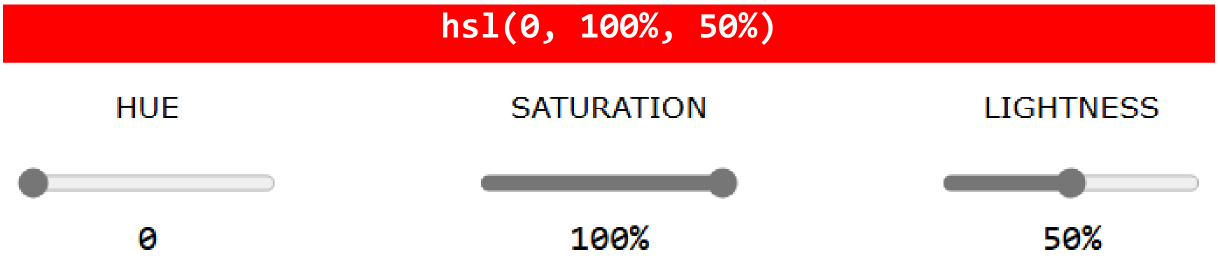
hsl(*hue, saturation, lightness*)

Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.

Saturation is a percentage value. 0% means a shade of gray, and 100% is the full color.

Lightness is also a percentage value. 0% is black, and 100% is white.

Experiment by mixing the HSL values below:



hsl(0, 100%, 50%)

HUE

0

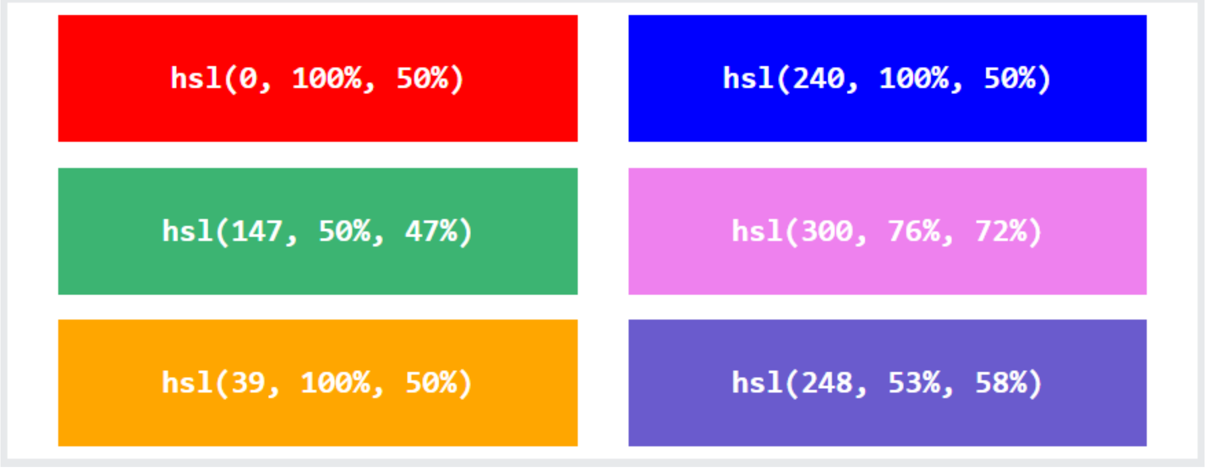
SATURATION

100%

LIGHTNESS

50%

Example



hsl(0, 100%, 50%)

hsl(240, 100%, 50%)

hsl(147, 50%, 47%)

hsl(300, 76%, 72%)

hsl(39, 100%, 50%)

hsl(248, 53%, 58%)

hsl(0, 100%, 50%)

hsl(240, 100%, 50%)

hsl(147, 50%, 47%)

hsl(300, 76%, 72%)

hsl(39, 100%, 50%)

hsl(248, 53%, 58%)

Try it Yourself »

Saturation

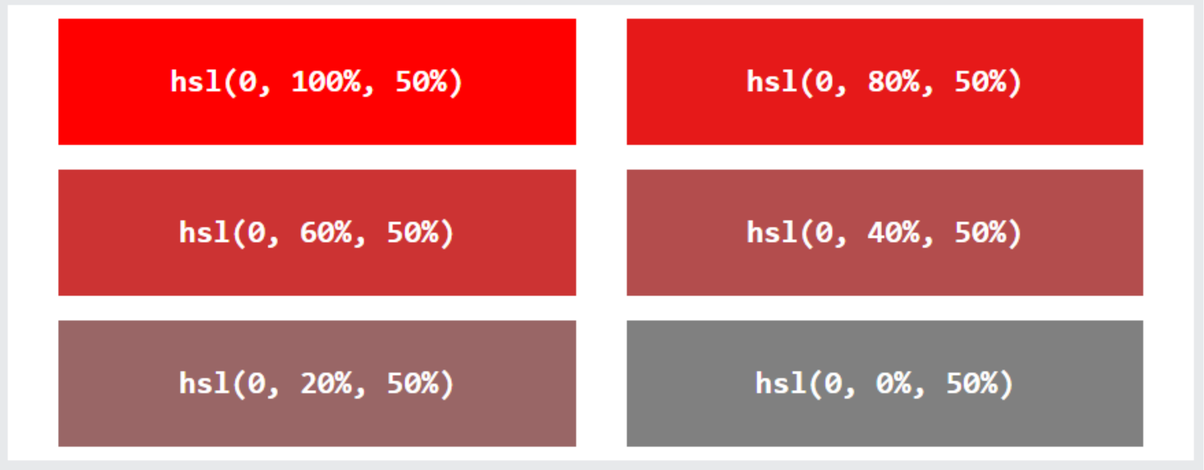
Saturation can be described as the intensity of a color.

100% is pure color, no shades of gray.

50% is 50% gray, but you can still see the color.

0% is completely gray; you can no longer see the color.

Example



hsl(0, 100%, 50%)
hsl(0, 80%, 50%)
hsl(0, 60%, 50%)
hsl(0, 40%, 50%)
hsl(0, 20%, 50%)
hsl(0, 0%, 50%)

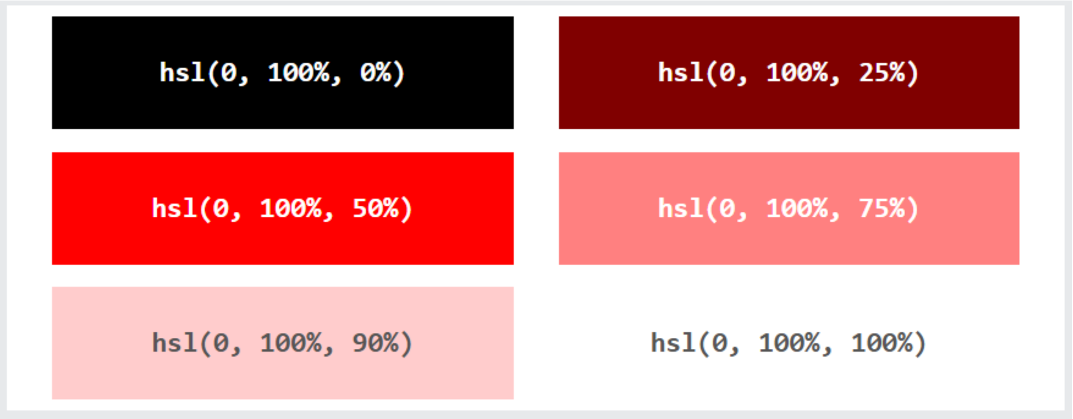
hsl(0, 100%, 50%)
hsl(0, 80%, 50%)
hsl(0, 60%, 50%)
hsl(0, 40%, 50%)
hsl(0, 20%, 50%)
hsl(0, 0%, 50%)

Try it Yourself »

Lightness

The lightness of a color can be described as how much light you want to give the color, where 0% means no light (black), 50% means 50% light (neither dark nor light), and 100% means full lightness (white).

Example



hsl(0, 100%, 0%)
hsl(0, 100%, 25%)
hsl(0, 100%, 50%)
hsl(0, 100%, 75%)
hsl(0, 100%, 90%)
hsl(0, 100%, 100%)

hsl(0, 100%, 0%)

hsl(0, 100%, 25%)

hsl(0, 100%, 50%)

hsl(0, 100%, 75%)

hsl(0, 100%, 90%)

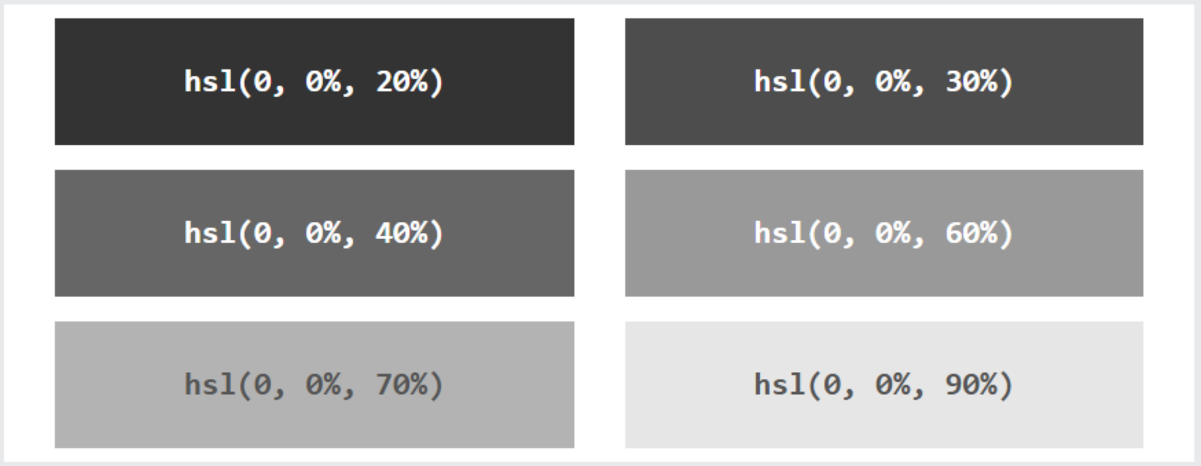
hsl(0, 100%, 100%)

Try it Yourself »

Shades of Gray

Shades of gray are often defined by setting the hue and saturation to 0, and adjusting the lightness from 0% to 100% to get darker/lighter shades:

Example



hsl(0, 0%, 20%)

hsl(0, 0%, 30%)

hsl(0, 0%, 40%)

hsl(0, 0%, 60%)

hsl(0, 0%, 70%)

hsl(0, 0%, 90%)

hsl(0, 0%, 20%)

hsl(0, 0%, 30%)

hsl(0, 0%, 40%)

hsl(0, 0%, 60%)

hsl(0, 0%, 70%)

hsl(0, 0%, 90%)

Try it Yourself »

HSLA Color Values

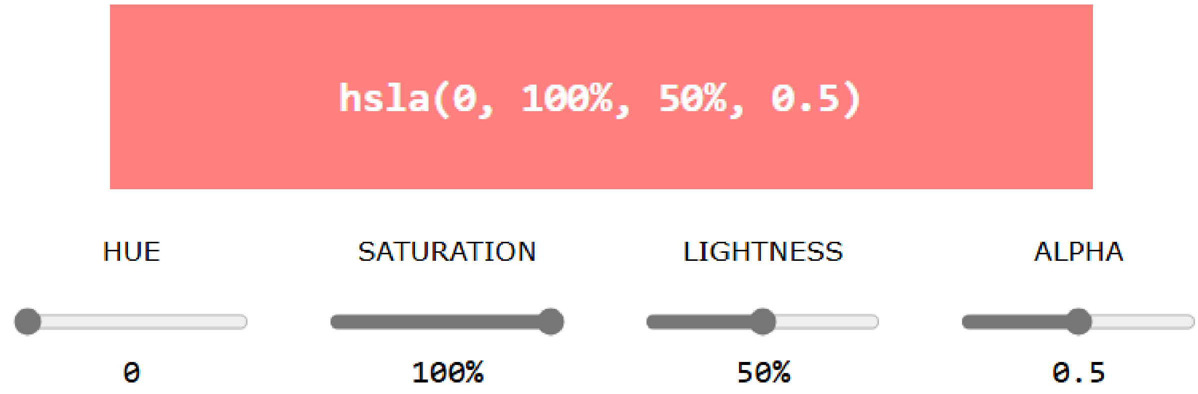
HSLA color values are an extension of HSL color values, with an Alpha channel - which specifies the opacity for a color.

An HSLA color value is specified with:

`hsla(`*`hue, saturation, lightness, alpha`*`)`

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

Experiment by mixing the HSLA values below:



hsla(0, 100%, 50%, 0.5)

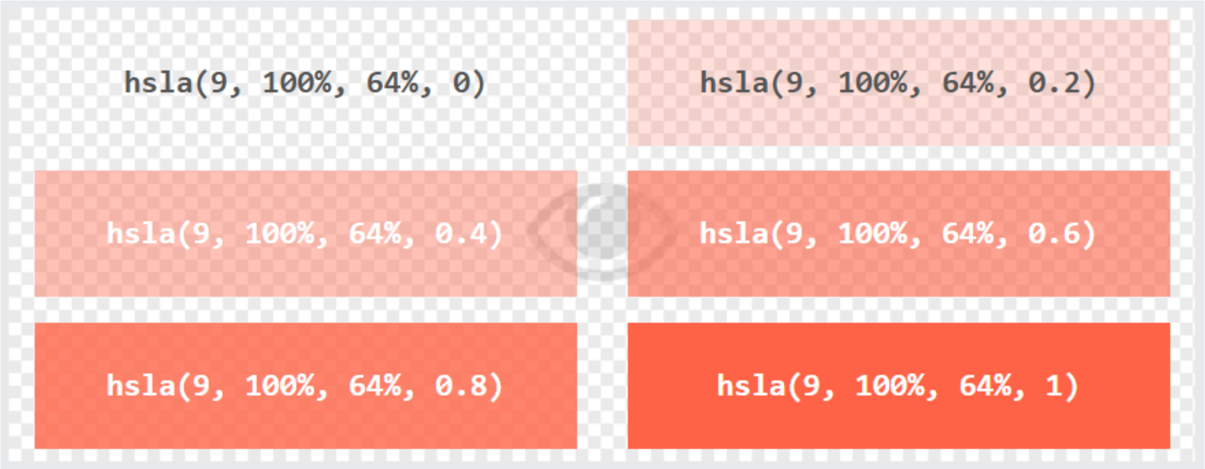
HUE
0

SATURATION
100%

LIGHTNESS
50%

ALPHA
0.5

Example



hsla(9, 100%, 64%, 0)
hsla(9, 100%, 64%, 0.2)
hsla(9, 100%, 64%, 0.4)
hsla(9, 100%, 64%, 0.6)
hsla(9, 100%, 64%, 0.8)
hsla(9, 100%, 64%, 1)

```
hsla(9, 100%, 64%, 0)
hsla(9, 100%, 64%, 0.2)
hsla(9, 100%, 64%, 0.4)
hsla(9, 100%, 64%, 0.6)
hsla(9, 100%, 64%, 0.8)
hsla(9, 100%, 64%, 1)
```

Try it Yourself »

HTML Exercises

Exercise:

Insert the HSLA value to make a color with no hue, 100% saturation, 50% lightness, and 50% transparency.

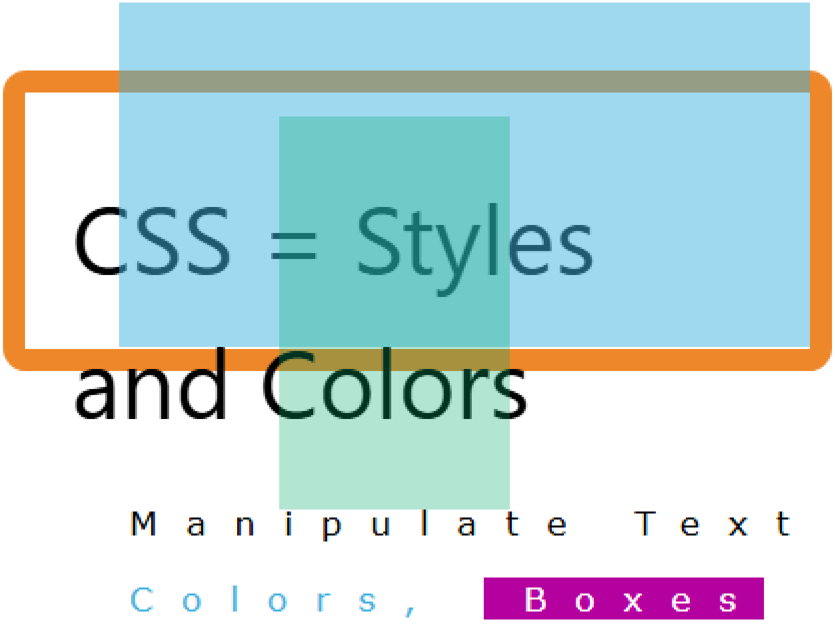
```
<p style="color: □ (□ )">This is a paragraph.</p>
```

Start the Exercise

HTML Styles - CSS

CSS stands for Cascading Style Sheets.

CSS saves a lot of work. It can control the layout of multiple web pages all at once.



What is CSS?

Cascading Style Sheets (CSS) is used to format the layout of a webpage.

With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!

Tip: The word **cascading** means that a style applied to a parent element will also apply to all children elements within the parent. So, if you set the color of the body text to "blue", all headings, paragraphs, and other text elements within the body will also get the same color (unless you specify something else)!

Using CSS

CSS can be added to HTML documents in 3 ways:

- **Inline** - by using the `style` attribute inside HTML elements
- **Internal** - by using a `<style>` element in the `<head>` section
- **External** - by using a `<link>` element to link to an external CSS file

The most common way to add CSS, is to keep the styles in external CSS files. However, in this tutorial we will use inline and internal styles, because this is easier to demonstrate, and easier for you to try it yourself.

Inline CSS

An inline CSS is used to apply a unique style to a single HTML element.

An inline CSS uses the `style` attribute of an HTML element.

The following example sets the text color of the `<h1>` element to blue, and the text color of the `<p>` element to red:

Example

```
<h1 style="color:blue;">A Blue Heading</h1>

<p style="color:red;">A red paragraph.</p>
```

Try it Yourself »

Internal CSS

An internal CSS is used to define a style for a single HTML page.

An internal CSS is defined in the `<head>` section of an HTML page, within a `<style>` element.

The following example sets the text color of ALL the `<h1>` elements (on that page) to blue, and the text color of ALL the `<p>` elements to red. In addition, the page will be displayed with a "powderblue" background color:

Example

```
<!DOCTYPE html>
<html>
<head>
```

```
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Try it Yourself »

External CSS

An external style sheet is used to define the style for many HTML pages.

To use an external style sheet, add a link to it in the `<head>` section of each HTML page:

Example

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Try it Yourself »

The external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension.

Here is what the "styles.css" file looks like:

"styles.css":

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

Tip: With an external style sheet, you can change the look of an entire web site, by changing one file!

CSS Colours, Fonts and Sizes

Here, we will demonstrate some commonly used CSS properties. You will learn more about them later.

The CSS `color` property defines the text colour to be used.

The CSS `font-family` property defines the font to be used.

The CSS `font-size` property defines the text size to be used.

Example

Use of CSS colour, font-family and font-size properties:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
p {
  color: red;
  font-family: courier;
  font-size: 160%;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

```
</body>
</html>
```

Try it Yourself »

CSS Border

The CSS `border` property defines a border around an HTML element.

Tip: You can define a border for nearly all HTML elements.

Example

Use of CSS border property:

```
p {
  border: 2px solid powderblue;
}
```

Try it Yourself »

CSS Padding

The CSS `padding` property defines a padding (space) between the text and the border.

Example

Use of CSS border and padding properties:

```
p {
  border: 2px solid powderblue;
  padding: 30px;
}
```

Try it Yourself »

CSS Margin

The CSS `margin` property defines a margin (space) outside the border.

Example

Use of CSS border and margin properties:

```
p {
  border: 2px solid powderblue;
  margin: 50px;
}
```

Try it Yourself »

Link to External CSS

External style sheets can be referenced with a full URL or with a path relative to the current web page.

Example

This example uses a full URL to link to a style sheet:

```
<link rel="stylesheet" href="https://www.w3schools.com/html/styles.css"
>
```

Try it Yourself »

Example

This example links to a style sheet located in the html folder on the current web site:

```
<link rel="stylesheet" href="/html/styles.css">
```

Try it Yourself »

Example

This example links to a style sheet located in the same folder as the current page:

```
<link rel="stylesheet" href="styles.css">
```

Try it Yourself »

You can read more about file paths in the chapter HTML File Paths.

Chapter Summary

- Use the HTML `style` attribute for inline styling
- Use the HTML `<style>` element to define internal CSS
- Use the HTML `<link>` element to refer to an external CSS file
- Use the HTML `<head>` element to store <style> and <link> elements
- Use the CSS `color` property for text colors
- Use the CSS `font-family` property for text fonts

- Use the CSS `font-size` property for text sizes
- Use the CSS `border` property for borders
- Use the CSS `padding` property for space inside the border
- Use the CSS `margin` property for space outside the border

Tip: You can learn much more about CSS in our CSS Tutorial.

HTML Exercises

Exercise:

Use CSS to set the background color of the document (body) to yellow.

```
<!DOCTYPE html>
<html>
<head>
<style>
□ □
  □:yellow;
□
</style>
</head>
<body>

<h1>My Home Page</h1>

</body>
</html>
```

Start the Exercise

HTML Style Tags

Tag	Description
<style>	Defines style information for an HTML document

<link>	Defines a link between a document and an external resource

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Links

Links are found in nearly all web pages. Links allow users to click their way from page to page.

HTML Links - Hyperlinks

HTML links are hyperlinks.

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

Note: A link does not have to be text. A link can be an image or any other HTML element!

HTML Links - Syntax

The HTML `<a>` tag defines a hyperlink. It has the following syntax:

```
<a href="url">link text</a>
```

The most important attribute of the `<a>` element is the `href` attribute, which indicates the link's destination.

The *link text* is the part that will be visible to the reader.

Clicking on the link text, will send the reader to the specified URL address.

Example

This example shows how to create a link to W3Schools.com:

```
<a href="https://www.w3schools.com/">Visit W3Schools.com!</a>
```

Try it Yourself »

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

Tip: Links can of course be styled with CSS, to get another look!

HTML Links - The target Attribute

By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link.

The `target` attribute specifies where to open the linked document.

The `target` attribute can have one of the following values:

- `_self` - Default. Opens the document in the same window/tab as it was clicked
- `_blank` - Opens the document in a new window or tab
- `_parent` - Opens the document in the parent frame
- `_top` - Opens the document in the full body of the window

Example

Use target="_blank" to open the linked document in a new browser window or tab:

```
<a href="https://www.w3schools.com/" target="_blank">Visit
W3Schools!</a>
```

Try it Yourself »

Absolute URLs vs. Relative URLs

Both examples above are using an **absolute URL** (a full web address) in the `href` attribute.

A local link (a link to a page within the same website) is specified with a **relative URL** (without the "https://www" part):

Example

```
<h2>Absolute URLs</h2>
<p><a href="https://www.w3.org/">W3C</a></p>
<p><a href="https://www.google.com/">Google</a></p>

<h2>Relative URLs</h2>
<p><a href="html_images.asp">HTML Images</a></p>
<p><a href="/css/default.asp">CSS Tutorial</a></p>
```

Try it Yourself »

HTML Links - Use an Image as a Link

To use an image as a link, just put the `<img>` tag inside the `<a>` tag:

Example

```
<a href="default.asp">
<img src="smiley.gif" alt="HTML
tutorial" style="width:42px;height:42px;">
</a>
```

Try it Yourself »

Link to an Email Address

Use `mailto:` inside the `href` attribute to create a link that opens the user's email program (to let them send a new email):

Example

```
<a href="mailto:someone@example.com">Send email</a>
```

Try it Yourself »

Button as a Link

To use an HTML button as a link, you have to add some JavaScript code.

JavaScript allows you to specify what happens at certain events, such as a click of a button:

Example

```
<button onclick="document.location='default.asp'">HTML
Tutorial</button>
```

Try it Yourself »

Tip: Learn more about JavaScript in our JavaScript Tutorial.

Link Titles

The `title` attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.

Example

```
<a href="https://www.w3schools.com/html/" title="Go to W3Schools HTML
section">Visit our HTML Tutorial</a>
```

Try it Yourself »

More on Absolute URLs and Relative URLs

Example

Use a full URL to link to a web page:

```
<a href="https://www.w3schools.com/html/default.asp">HTML tutorial</a>
```

Try it Yourself »

Example

Link to a page located in the html folder on the current web site:

```
<a href="/html/default.asp">HTML tutorial</a>
```

Try it Yourself »

Example

Link to a page located in the same folder as the current page:

```
<a href="default.asp">HTML tutorial</a>
```

Try it Yourself »

You can read more about file paths in the chapter HTML File Paths.

Chapter Summary

- Use the `<a>` element to define a link
- Use the `href` attribute to define the link address
- Use the `target` attribute to define where to open the linked document
- Use the `<img>` element (inside `<a>`) to use an image as a link
- Use the `mailto:` scheme inside the `href` attribute to create a link that opens the user's email program

HTML Link Tags

Tag	Description
<a>	Defines a hyperlink

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Exercises

Exercise:

Use the correct HTML to make the text below into a link to "default.html".

```
□ □>Visit our HTML tutorial.□
```

Start the Exercise

HTML Links - Different Colors

An HTML link is displayed in a different color depending on whether it has been visited, is unvisited, or is active.

HTML Link Colors

By default, a link will appear like this (in all browsers):

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

You can change the link state colors, by using CSS:

Example

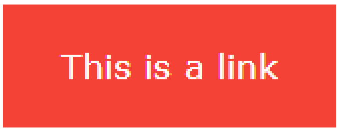
Here, an unvisited link will be green with no underline. A visited link will be pink with no underline. An active link will be yellow and underlined. In addition, when mousing over a link (a:hover) it will become red and underlined:

```
<style>
a:link {
  color: green;
  background-color: transparent;
  text-decoration: none;
}

a:visited {
  color: pink;
  background-color: transparent;
  text-decoration: none;
}

a:hover {
  color: red;
  background-color: transparent;
  text-decoration: underline;
}

a:active {
  color: yellow;
  background-color: transparent;
  text-decoration: underline;
}
</style>
```

Try it Yourself »

Link Buttons

A link can also be styled as a button, by using CSS:



This is a link

Example

```
<style>
a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 15px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

a:hover, a:active {
  background-color: red;
}
</style>
```

Try it Yourself »

To learn more about CSS, go to our CSS Tutorial.

HTML Link Tags

Tag	Description
<a>	Defines a hyperlink

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Exercises

Exercise:

Use CSS to remove the underline from the link.

```
<a href="html_images.asp" style="□">HTML Images</a>
```

Start the Exercise

HTML Links - Create Bookmarks

HTML links can be used to create bookmarks, so that readers can jump to specific parts of a web page.

Create a Bookmark in HTML

Bookmarks can be useful if a web page is very long.

To create a bookmark - first create the bookmark, then add a link to it.

When the link is clicked, the page will scroll down or up to the location with the bookmark.

Example

First, use the id attribute to create a bookmark:

```
<h2 id="C4">Chapter 4</h2>
```

Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:

Example

```
<a href="#C4">Jump to Chapter 4</a>
```

Try it Yourself »

You can also add a link to a bookmark on another page:

```
<a href="html_demo.html#C4">Jump to Chapter 4</a>
```

Chapter Summary

- Use the id attribute (id="*value*") to define bookmarks in a page
- Use the href attribute (href="#*value*") to link to the bookmark

HTML Exercises

Exercise:

Add a link to a bookmark, on the same page, with the id="mytext".

```
<a □="□">Jump to new text</a>
```

Start the Exercise

HTML Link Tags

Tag	Description
<a>	Defines a hyperlink

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Images

Images can improve the design and the appearance of a web page.

Example

```
<img src="pic_trulli.jpg" alt="Italian Trulli">
```

Try it Yourself »

Example

```
<img src="img_girl.jpg" alt="Girl in a jacket">
```

Try it Yourself »

Example

```
<img src="img_chania.jpg" alt="Flowers in Chania">
```

Try it Yourself »

HTML Images Syntax

The HTML `<img>` tag is used to embed an image in a web page.

Images are not technically inserted into a web page; images are linked to web pages. The `<img>` tag creates a holding space for the referenced image.

The `<img>` tag is empty, it contains attributes only, and does not have a closing tag.

The `<img>` tag has two required attributes:

- src - Specifies the path to the image
- alt - Specifies an alternate text for the image

Syntax

```
<img src="url" alt="alternatetext">
```

The src Attribute

The required `src` attribute specifies the path (URL) to the image.

Note: When a web page loads, it is the browser, at that moment, that gets the image from a web server and inserts it into the page. Therefore, make sure that the image actually stays in the same spot in relation to the web page, otherwise your visitors will get a broken link icon. The broken link icon and the alt text are shown if the browser cannot find the image.

Example

```
<img src="img_chania.jpg" alt="Flowers in Chania">
```

Try it Yourself »

The alt Attribute

The required alt attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader).

The value of the alt attribute should describe the image:

Example

```
<img src="img_chania.jpg" alt="Flowers in Chania">
```

Try it Yourself »

If a browser cannot find an image, it will display the value of the alt attribute:

Example

```
<img src="wrongname.gif" alt="Flowers in Chania">
```

Try it Yourself »

Tip: A screen reader is a software program that reads the HTML code, and allows the user to "listen" to the content. Screen readers are useful for people who are visually impaired or learning disabled.

Image Size - Width and Height

You can use the style attribute to specify the width and height of an image.

Example

```
<img src="img_girl.jpg" alt="Girl in a
jacket" style="width:500px;height:600px;">
```

Try it Yourself »

Alternatively, you can use the width and height attributes:

Example

```
<img src="img_girl.jpg" alt="Girl in a
jacket" width="500" height="600">
```

Try it Yourself »

The width and height attributes always define the width and height of the image in pixels.

Note: Always specify the width and height of an image. If width and height are not specified, the web page might flicker while the image loads.

Width and Height, or Style?

The width, height, and style attributes are all valid in HTML.

However, we suggest using the style attribute. It prevents styles sheets from changing the size of images:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  width: 100%;
}
</style>
</head>
<body>

<img src="html5.gif" alt="HTML5 Icon" width="128" height="128">

<img src="html5.gif" alt="HTML5
Icon" style="width:128px;height:128px;">

</body>
</html>
```

Try it Yourself »

Images in Another Folder

If you have your images in a sub-folder, you must include the folder name in the src attribute:

Example

```
<img src="/images/html5.gif" alt="HTML5
Icon" style="width:128px;height:128px;">
```

Try it Yourself »

Images on Another Server/Website

Some web sites point to an image on another server.

To point to an image on another server, you must specify an absolute (full) URL in the src attribute:

Example

```
<img src="https://www.w3schools.com/images/w3schools_green.jpg" alt="W3
Schools.com">
```

Try it Yourself »

Notes on external images: External images might be under copyright. If you do not get permission to use it, you may be in violation of copyright laws. In addition, you cannot control external images; they can suddenly be removed or changed.

Animated Images

HTML allows animated GIFs:

Example

```
<img src="programming.gif" alt="Computer
Man" style="width:48px;height:48px;">
```

Try it Yourself »

Image as a Link

To use an image as a link, put the <img> tag inside the <a> tag:

Example

```
<a href="default.asp">
  <img src="smiley.gif" alt="HTML
```

```
tutorial" style="width:42px;height:42px;">
</a>
```

Try it Yourself »

Image Floating

Use the CSS `float` property to let the image float to the right or to the left of a text:

Example

```
<p><img src="smiley.gif" alt="Smiley
face" style="float:right;width:42px;height:42px;">
The image will float to the right of the text.</p>

<p><img src="smiley.gif" alt="Smiley
face" style="float:left;width:42px;height:42px;">
The image will float to the left of the text.</p>
```

Try it Yourself »

Tip: To learn more about CSS Float, read our CSS Float Tutorial.

Common Image Formats

Here are the most common image file types, which are supported in all browsers (Chrome, Edge, Firefox, Safari, Opera):

Abbreviation	File Format	File Extension
APNG	Animated Portable Network Graphics	.apng
GIF	Graphics Interchange Format	.gif
ICO	Microsoft Icon	.ico, .cur

JPEG	Joint Photographic Expert Group image	.jpg, .jpeg, .jfif, .pjpeg, .pjp
PNG	Portable Network Graphics	.png
SVG	Scalable Vector Graphics	.svg

Chapter Summary

- Use the HTML `<img>` element to define an image
- Use the HTML `src` attribute to define the URL of the image
- Use the HTML `alt` attribute to define an alternate text for an image, if it cannot be displayed
- Use the HTML `width` and `height` attributes or the CSS `width` and `height` properties to define the size of the image
- Use the CSS `float` property to let the image float to the left or to the right

Note: Loading large images takes time, and can slow down your web page. Use images carefully.

HTML Exercises

Exercise:

Use the HTML image attributes to set the size of the image to 250 pixels wide and 400 pixels tall.

```
<img src="scream.png" □="250" □="400">
```

Start the Exercise

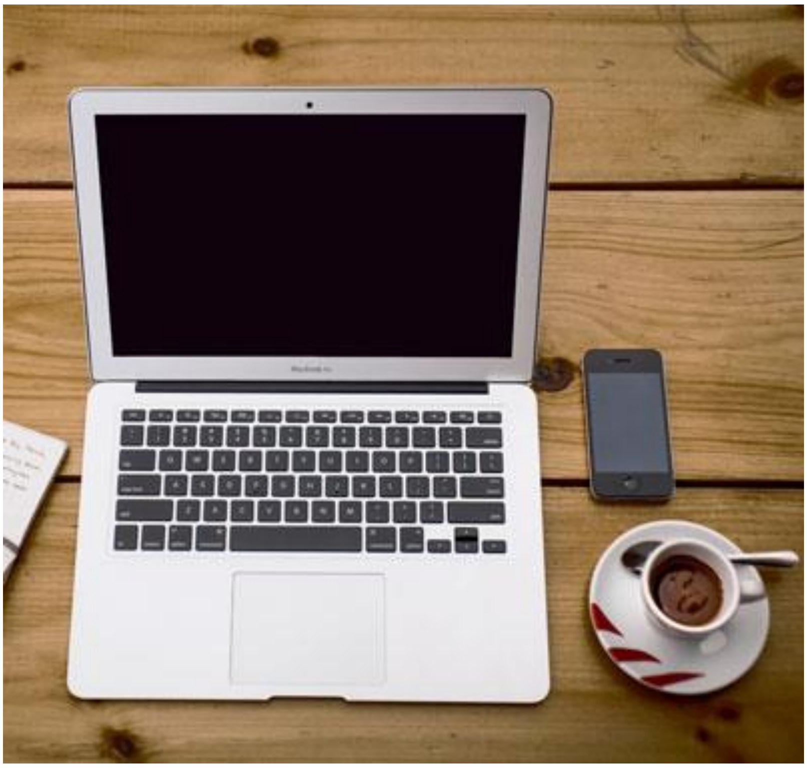
HTML Image Maps

With HTML image maps, you can create clickable areas on an image.

Image Maps

The HTML `<map>` tag defines an image map. An image map is an image with clickable areas. The areas are defined with one or more `<area>` tags.

Try to click on the computer, phone, or the cup of coffee in the image below:



Example

Here is the HTML source code for the image map above:

```
<img src="workplace.jpg" alt="Workplace" usemap="#workmap">

<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">
</map>
```

Try it Yourself »

How Does it Work?

The idea behind an image map is that you should be able to perform different actions depending on where in the image you click.

To create an image map you need an image, and some HTML code that describes the clickable areas.

The Image

The image is inserted using the `<img>` tag. The only difference from other images is that you must add a `usemap` attribute:

```
<img src="workplace.jpg" alt="Workplace" usemap="#workmap">
```

The `usemap` value starts with a hash tag `#` followed by the name of the image map, and is used to create a relationship between the image and the image map.

Tip: You can use any image as an image map!

Create Image Map

Then, add a `<map>` element.

The `<map>` element is used to create an image map, and is linked to the image by using the required `name` attribute:

```
<map name="workmap">
```

The `name` attribute must have the same value as the `<img>`'s `usemap` attribute .

The Areas

Then, add the clickable areas.

A clickable area is defined using an `<area>` element.

Shape

You must define the shape of the clickable area, and you can choose one of these values:

- `rect` - defines a rectangular region
- `circle` - defines a circular region
- `poly` - defines a polygonal region
- `default` - defines the entire region

You must also define some coordinates to be able to place the clickable area onto the image.

Shape="rect"

The coordinates for `shape="rect"` come in pairs, one for the x-axis and one for the y-axis.

So, the coordinates `34,44` is located 34 pixels from the left margin and 44 pixels from the top:

The coordinates `270,350` is located 270 pixels from the left margin and 350 pixels from the top:

Now we have enough data to create a clickable rectangular area:

Example

```
<area shape="rect" coords="34, 44, 270, 350" href="computer.htm">
```

Try it Yourself »

This is the area that becomes clickable and will send the user to the page "computer.htm":

Shape="circle"

To add a circle area, first locate the coordinates of the center of the circle:

`337,300`

Then specify the radius of the circle:

44 pixels

Now you have enough data to create a clickable circular area:

Example

```
<area shape="circle" coords="337, 300, 44" href="coffee.htm">
```

Try it Yourself »

This is the area that becomes clickable and will send the user to the page "coffee.htm":

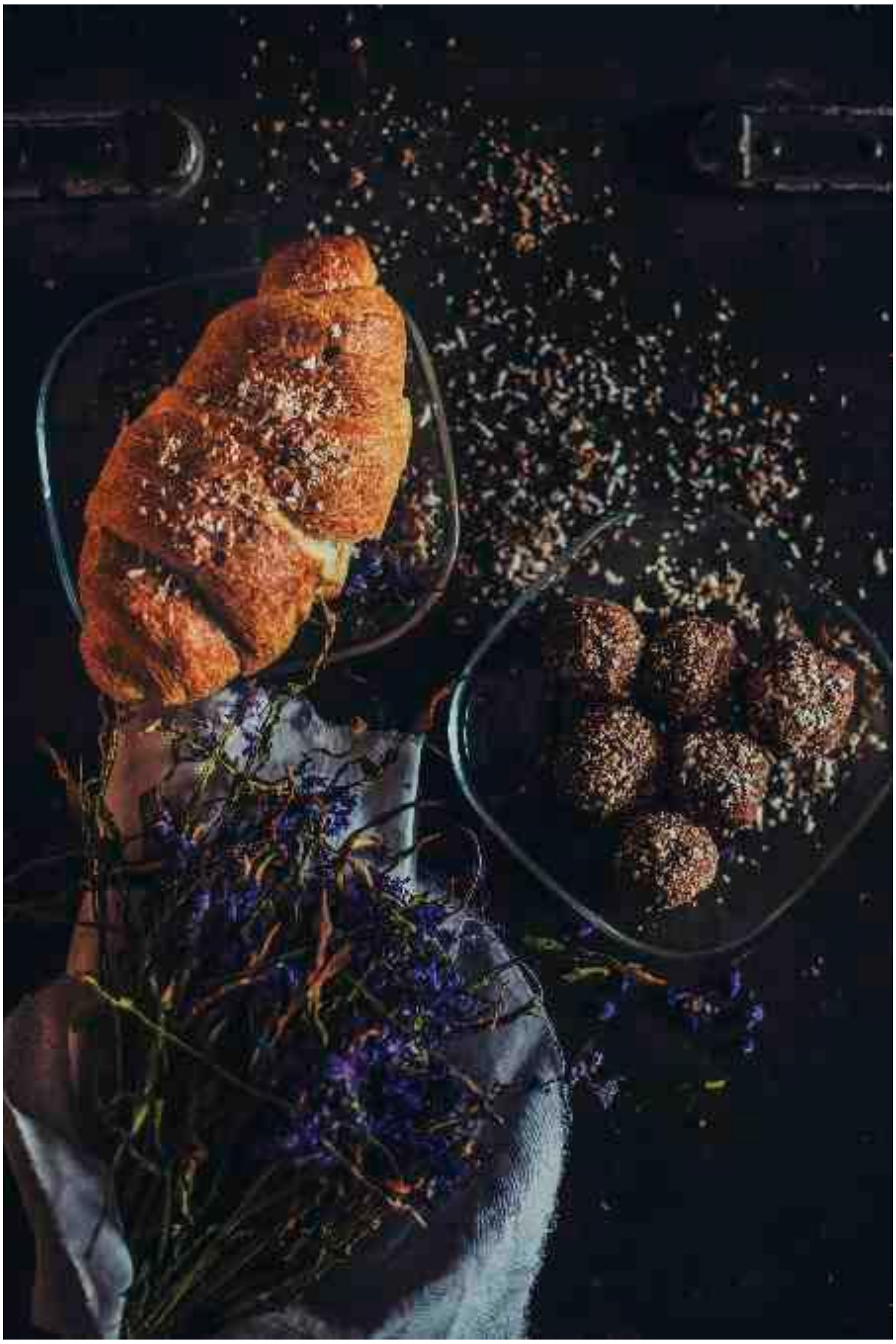
Shape="poly"

The `shape="poly"` contains several coordinate points, which creates a shape formed with straight lines (a polygon).

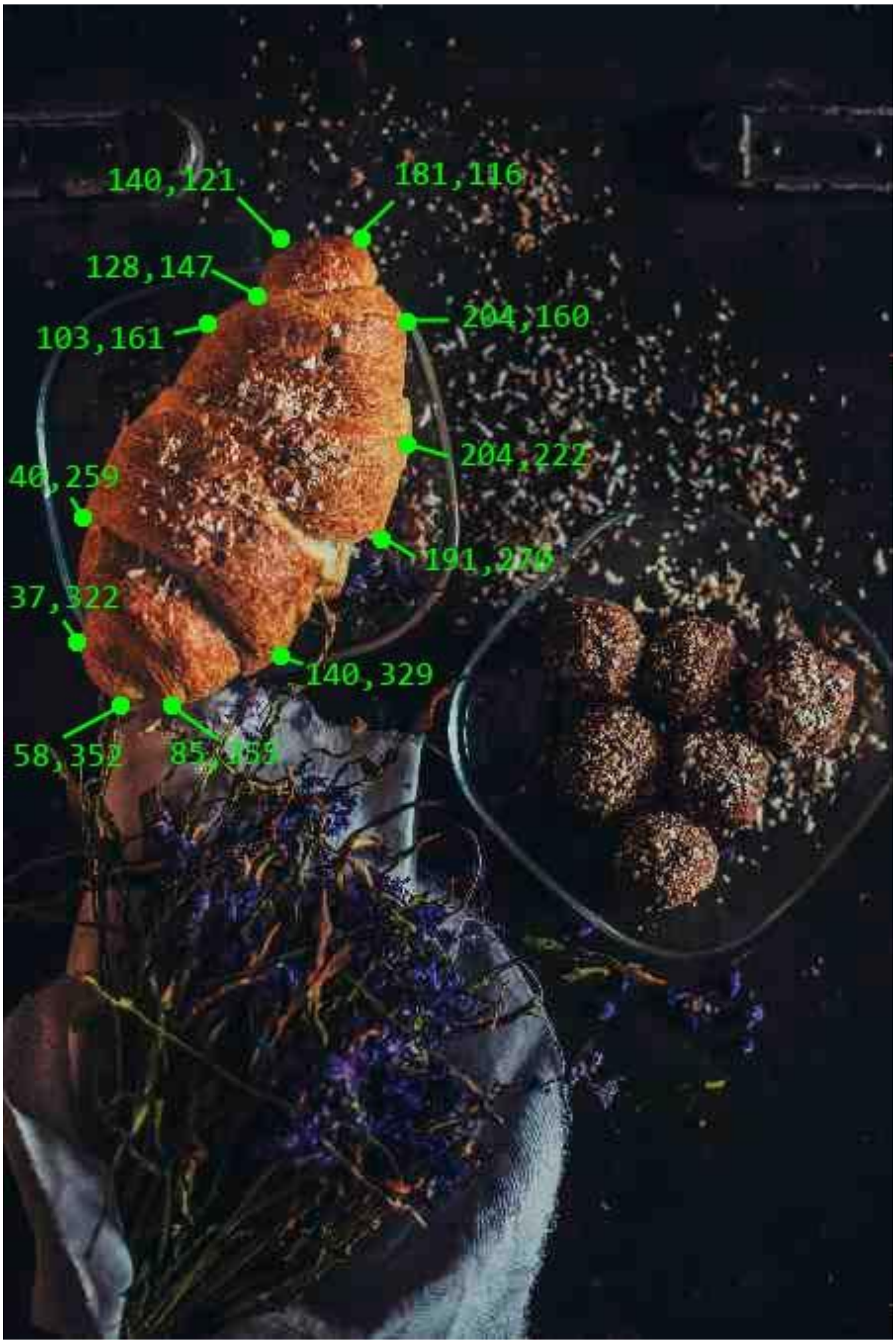
This can be used to create any shape.

Like maybe a croissant shape!

How can we make the croissant in the image below become a clickable link?



We have to find the x and y coordinates for all edges of the croissant:



The coordinates come in pairs, one for the x-axis and one for the y-axis:

Example

```
<area shape="poly" coords="140,121,181,116,204,160,204,222,191,270,140,
329,85,355,58,352,37,322,40,259,103,161,128,147" href="croissant.htm">
```

Try it Yourself »

This is the area that becomes clickable and will send the user to the page "croissant.htm":

Image Map and JavaScript

A clickable area can also trigger a JavaScript function.

Add a `click` event to the `<area>` element to execute a JavaScript function:

Example

Here, we use the onclick attribute to execute a JavaScript function when the area is clicked:

```
<map name="workmap">
  <area shape="circle" coords="337,300,44" href="coffee.htm" onclick="myFunction()">
</map>

<script>
function myFunction() {
  alert("You clicked the coffee cup!");
}
</script>
```

Try it Yourself »

Chapter Summary

- Use the HTML `<map>` element to define an image map
- Use the HTML `<area>` element to define the clickable areas in the image map
- Use the HTML `usemap` attribute of the `<img>` element to point to an image map

HTML Exercises

Exercise:

For image maps, specify a legal value for the area tag's shape attribute.

(Any legal value will give a correct answer)

`<area shape="□" href="default.html">`

Start the Exercise

HTML Image Tags

Tag	Description
<img>	Defines an image
<map>	Defines an image map

<area>	Defines a clickable area inside an image map
<picture>	Defines a container for multiple image resources

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Background Images

A background image can be specified for almost any HTML element.

Background Image on a HTML element

To add a background image on an HTML element, use the HTML `style` attribute and the CSS `background-image` property:

Example

Add a background image on a HTML element:

```
<p style="background-image: url('img_girl.jpg');">
```

Try it Yourself »

You can also specify the background image in the `<style>` element, in the `<head>` section:

Example

Specify the background image in the `<style>` element:

```
<style>
p {
  background-image: url('img_girl.jpg');
}
</style>
```

Try it Yourself »

Background Image on a Page

If you want the entire page to have a background image, you must specify the background image on the `<body>` element:

Example

Add a background image for the entire page:

```
<style>
body {
  background-image: url('img_girl.jpg');
}
</style>
```

Try it Yourself »

Background Repeat

If the background image is smaller than the element, the image will repeat itself, horizontally and vertically, until it reaches the end of the element:

Example

```
<style>
body {
  background-image: url('example_img_girl.jpg');
}
</style>
```

Try it Yourself »

To avoid the background image from repeating itself, set the `background-repeat` property to `no-repeat`.

Example

```
<style>
body {
  background-image: url('example_img_girl.jpg');
  background-repeat: no-repeat;
}
</style>
```

Try it Yourself »

Background Cover

If you want the background image to cover the entire element, you can set the `background-size` property to `cover.`

Also, to make sure the entire element is always covered, set the `background-attachment` property to `fixed:`

This way, the background image will cover the entire element, with no stretching (the image will keep its original proportions):

Example

```
<style>
body {
  background-image: url('img_girl.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
}
</style>
```

Try it Yourself »

Background Stretch

If you want the background image to stretch to fit the entire element, you can set the `background-size` property to `100% 100%`:

Try resizing the browser window, and you will see that the image will stretch, but always cover the entire element.

Example

```
<style>
body {
  background-image: url('img_girl.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: 100% 100%;
}
</style>
```

Try it Yourself »

HTML Exercises

Exercise:

Add a background image named "myimage.png" on a HTML element.

```
<p style="background-image: □">
```

Start the Exercise

Learn More CSS

From the examples above you have learned that background images can be styled by using the CSS background properties.

To learn more about CSS background properties, study our CSS Background Tutorial.

HTML <picture> Element

The HTML `<picture>` element allows you to display different pictures for different devices or screen sizes.

The HTML <picture> Element

The HTML `<picture>` element gives web developers more flexibility in specifying image resources.

The `<picture>` element contains one or more `<source>` elements, each referring to different images through the `srcset` attribute. This way the browser can choose the image that best fits the current view and/or device.

Each `<source>` element has a `media` attribute that defines when the image is the most suitable.

Example

Show different images for different screen sizes:

```
<picture>
  <source media="(min-width: 650px)" srcset="img_food.jpg">
  <source media="(min-width: 465px)" srcset="img_car.jpg">
  <img src="img_girl.jpg">
</picture>
```

Try it Yourself »

Note: Always specify an <img> element as the last child element of the <picture> element. The <img> element is used by browsers that do not support the <picture> element, or if none of the <source> tags match.

When to use the Picture Element

There are two main purposes for the <picture> element:

1. Bandwidth

If you have a small screen or device, it is not necessary to load a large image file. The browser will use the first <source> element with matching attribute values, and ignore any of the following elements.

2. Format Support

Some browsers or devices may not support all image formats. By using the <picture> element, you can add images of all formats, and the browser will use the first format it recognizes, and ignore any of the following elements.

Example

The browser will use the first image format it recognizes:

```
<picture>
  <source srcset="img_avatar.png">
  <source srcset="img_girl.jpg">
  <img src="img_beatles.gif" alt="Beatles" style="width:auto;">
</picture>
```

Try it Yourself »

Note: The browser will use the first <source> element with matching attribute values, and ignore any following <source> elements.

HTML Image Tags

Tag	Description
<img>	Defines an image

<map>	Defines an image map
<area>	Defines a clickable area inside an image map
<picture>	Defines a container for multiple image resources

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Favicon

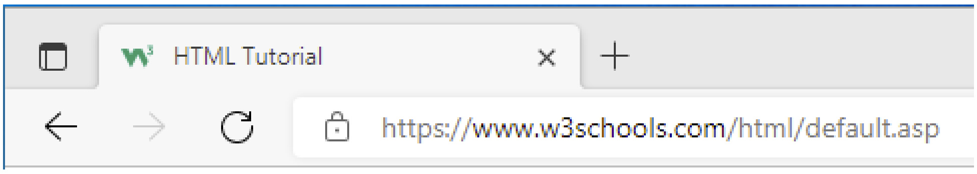
A favicon is a small image displayed next to the page title in the browser tab.

How To Add a Favicon in HTML

You can use any image you like as your favicon. You can also create your own favicon on sites like https://www.favicon.cc.

Tip: A favicon is a small image, so it should be a simple image with high contrast.

A favicon image is displayed to the left of the page title in the browser tab, like this:



To add a favicon to your website, either save your favicon image to the root directory of your webserver, or create a folder in the root directory called images, and save your favicon image in this folder. A common name for a favicon image is "favicon.ico".

Next, add a `<link>` element to your "index.html" file, after the `<title>` element, like this:

Example

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page Title</title>
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico">
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Now, save the "index.html" file and reload it in your browser. Your browser tab should now display your favicon image to the left of the page title.

Favicon File Format Support

The following table shows the file format support for a favicon image:

Browser	**ICO**	**PNG**	**GIF**	**JPEG**	**SVG**
Edge	Yes	Yes	Yes	Yes	Yes
Chrome	Yes	Yes	Yes	Yes	Yes
Firefox	Yes	Yes	Yes	Yes	Yes
Opera	Yes	Yes	Yes	Yes	Yes
Safari	Yes	Yes	Yes	Yes	Yes

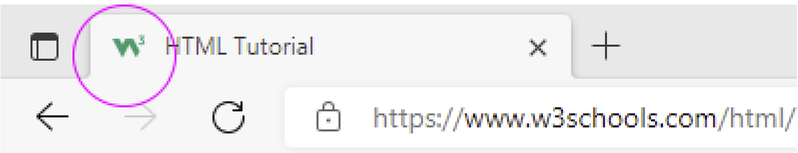
Chapter Summary

- Use the HTML `<link>` element to insert a favicon

HTML Exercises

Exercise:

Which HTML tag is used to handle the favicon (the little image to the left in the browser tab):



```
<□ rel="icon" type="image/x-icon"
href="/images/favicon.ico">
```

[Start the Exercise](#)

HTML Link Tag

Tag	Description
<link>	Defines the relationship between a document and an external resource

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Page Title

Every web page should have a page title to describe the meaning of the page.

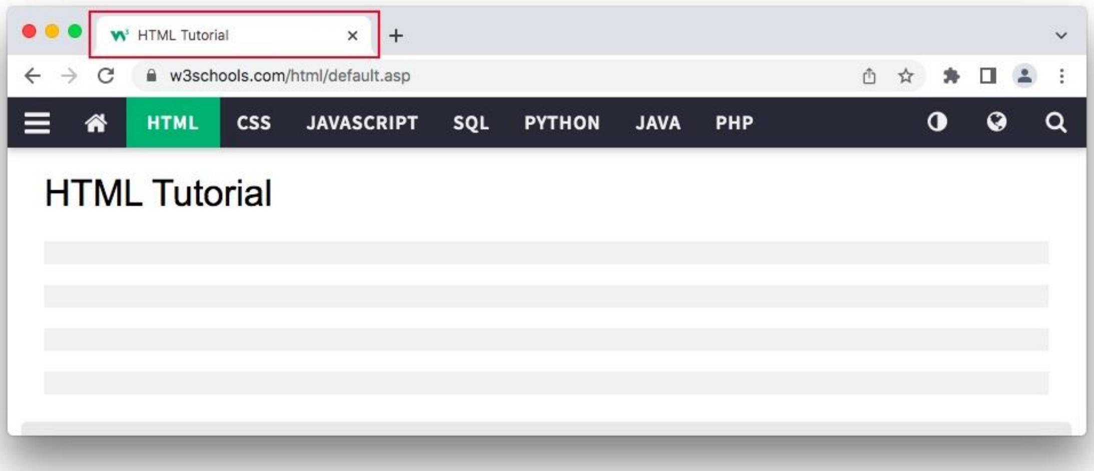
The `<title>` element adds a title to your page:

Example

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Tutorial</title>
</head>
<body>
```

```
The content of the document......

</body>
</html>
```

The title is shown in the browser's title bar:



The title should describe the content and the meaning of the page.

The page title is very important for search engine optimization (SEO). The text is used by search engine algorithms to decide the order when listing pages in search results.

The `<title>` element:

- defines a title in the browser toolbar
- provides a title for the page when it is added to favorites
- displays a title for the page in search engine-results

So, try to make the title as accurate and meaningful as possible!

HTML Title Tag

Tag	Description
<title>	Defines the title of the document

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Tables

HTML tables allow web developers to arrange data into rows and columns.

Example

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

Try it Yourself »

Define an HTML Table

A table in HTML consists of table cells inside rows and columns.

Example

A simple HTML table:

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>
```

Try it Yourself »

Table Cells

Each table cell is defined by a `<td>` and a `</td>` tag.

`td` stands for table data.

Everything between `<td>` and `</td>` are the content of the table cell.

Example

```
<table>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
</table>
```

Try it Yourself »

Note: A table cell can contain all sorts of HTML elements: text, images, lists, links, other tables, etc.

Table Rows

Each table row starts with a `<tr>` and ends with a `</tr>` tag.

`tr` stands for table row.

Example

```
<table>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
  </tr>
</table>
```

Try it Yourself »

You can have as many rows as you like in a table; just make sure that the number of cells are the same in each row.

Note: There are times when a row can have less or more cells than another. You will learn about that in a later chapter.

Table Headers

Sometimes you want your cells to be table header cells. In those cases use the `<th>` tag instead of the `<td>` tag:

`th` stands for table header.

Example

Let the first row be table header cells:

```
<table>
  <tr>
    <th>Person 1</th>
    <th>Person 2</th>
    <th>Person 3</th>
  </tr>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
  </tr>
</table>
```

Try it Yourself »

By default, the text in `<th>` elements are bold and centered, but you can change that with CSS.

HTML Exercises

Exercise:

Add a table row with two table headers.

The two table headers should have the value "Name" and "Age".

```
<table>
  □
    □
    □
  □
  <tr>
    <td>Jill Smith</td>
    <td>50</td>
  </tr>
</table>
```

Start the Exercise

HTML Table Tags

Tag	Description
<table>	Defines a table
<th>	Defines a header cell in a table

<tr>	Defines a row in a table
<td>	Defines a cell in a table
<caption>	Defines a table caption
<colgroup>	Specifies a group of one or more columns in a table for formatting
<col>	Specifies column properties for each column within a <colgroup> element
<thead>	Groups the header content in a table
<tbody>	Groups the body content in a table
<tfoot>	Groups the footer content in a table

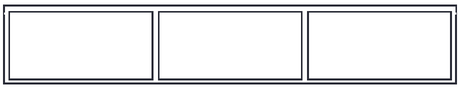
For a complete list of all available HTML tags, visit our HTML Tag Reference.

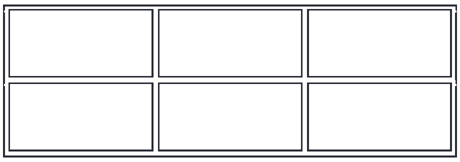
HTML Table Borders

HTML tables can have borders of different styles and shapes.

How To Add a Border

To add a border, use the CSS `border` property on `table`, `th`, and `td` elements:





Example

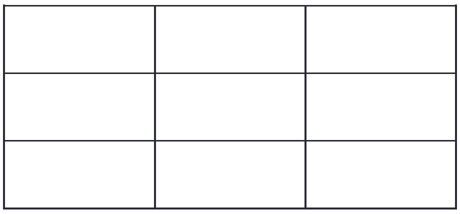
```
table, th, td {
  border: 1px solid black;
}
```

Try it Yourself »

Collapsed Table Borders

To avoid having double borders like in the example above, set the CSS `border-collapse` property to `collapse`.

This will make the borders collapse into a single border:



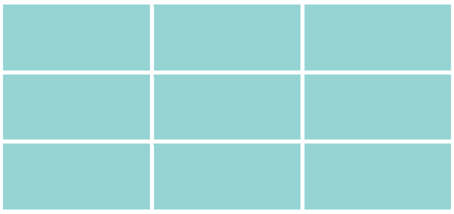
Example

```
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
```

Try it Yourself »

Style Table Borders

If you set a background color of each cell, and give the border a white color (the same as the document background), you get the impression of an invisible border:



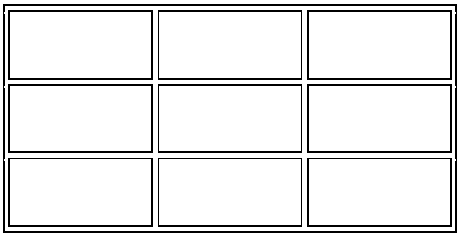
Example

```
table, th, td {
  border: 1px solid white;
  border-collapse: collapse;
}
th, td {
  background-color: #96D4D4;
}
```

Try it Yourself »

Round Table Borders

With the `border-radius` property, the borders get rounded corners:

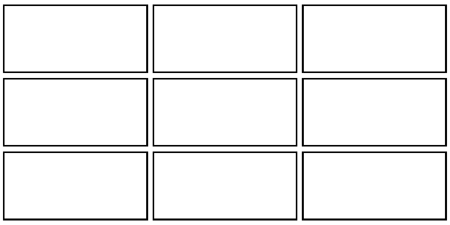


Example

```
table, th, td {
  border: 1px solid black;
  border-radius: 10px;
}
```

Try it Yourself »

Skip the border around the table by leaving out `table` from the css selector:



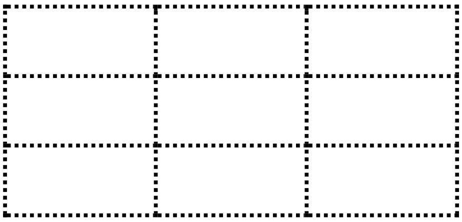
Example

```
th, td {
  border: 1px solid black;
  border-radius: 10px;
}
```

Try it Yourself »

Dotted Table Borders

With the `border-style` property, you can set the appearance of the border.



The following values are allowed:

- `dotted`
- `dashed`
- `solid`
- `double`
- `groove`
- `ridge`
- `inset`
- `outset`
- `none`
- `hidden`

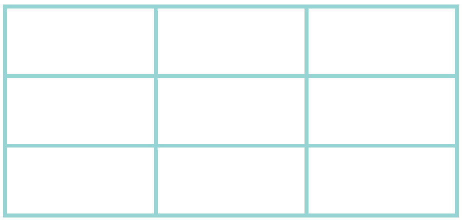
Example

```
 th, td {
  border-style: dotted;
}
```

Try it Yourself »

Border Colour

With the `border-color` property, you can set the color of the border.



Example

```
 th, td {
  border-color: #96D4D4;
}
```

Try it Yourself »

HTML Exercises

Exercise:

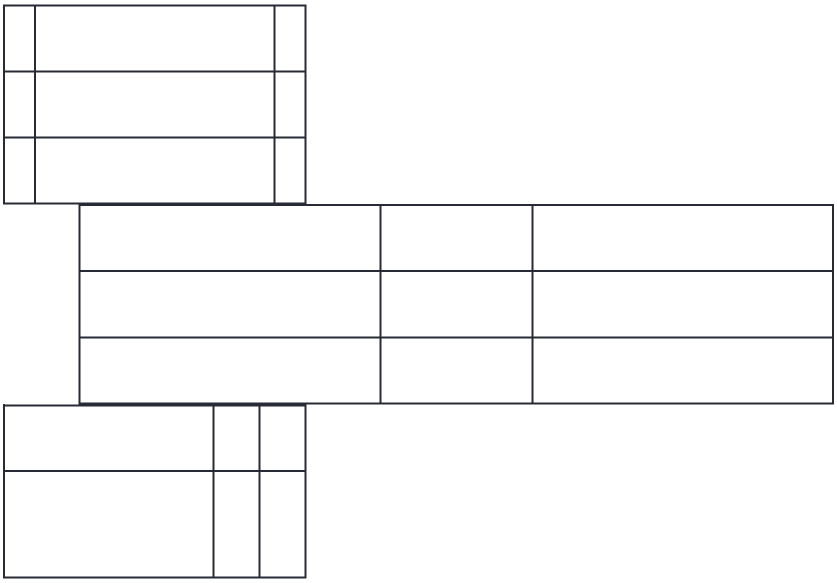
Use the correct CSS border values to create a solid black 3 pixels border on a table element.

```
table, th, td {
  border: □;
}
```

Start the Exercise

HTML Table Sizes

HTML tables can have different sizes for each column, row or the entire table.



Use the `style` attribute with the `width` or `height` properties to specify the size of a table, row or column.

HTML Table Width

To set the width of a table, add the `style` attribute to the `<table>` element:

Example

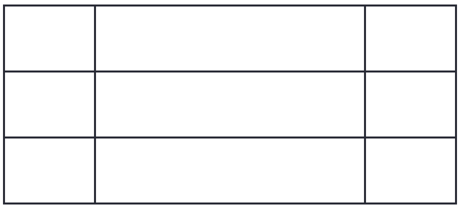
Set the width of the table to 100%:

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

Try it Yourself »

Note: Using a percentage as the size unit for a width means how wide will this element be compared to its parent element, which in this case is the `<body>` element.

HTML Table Column Width



To set the size of a specific column, add the `style` attribute on a `<th>` or `<td>` element:

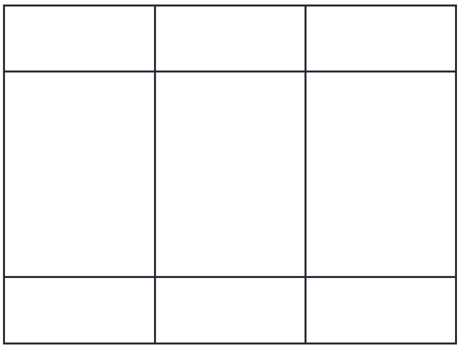
Example

Set the width of the first column to 70%:

```
<table style="width:100%">
  <tr>
    <th style="width:70%">Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

Try it Yourself »

HTML Table Row Height



To set the height of a specific row, add the `style` attribute on a table row element:

Example

Set the height of the second row to 200 pixels:

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr style="height:200px">
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

Try it Yourself »

HTML Exercises

Exercise:

Use CSS styles to make the table 300 pixels wide.

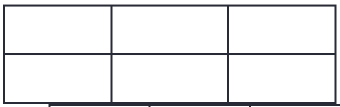
```
<table       >
  <tr>
    <th>First Name</th>
    <th>Last Name</th>
    <th>Points</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
</table>
```

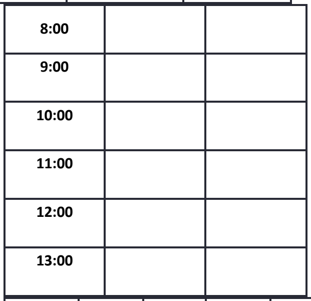
Start the Exercise

HTML Table Headers

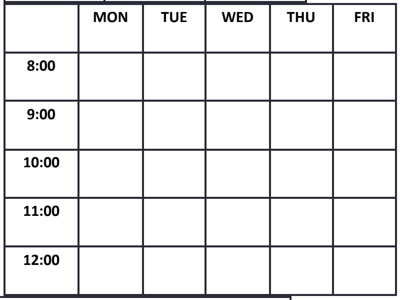
HTML tables can have headers for each column or row, or for many columns/rows.

EMIL	TOBIAS	LINUS

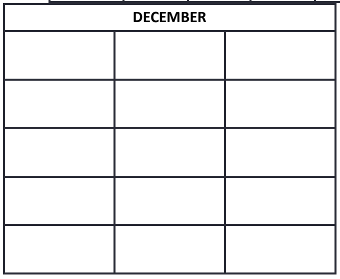




8:00		
9:00		
10:00		
11:00		
12:00		
13:00		



	MON	TUE	WED	THU	FRI
8:00					
9:00					
10:00					
11:00					
12:00					



DECEMBER		

HTML Table Headers

Table headers are defined with th elements. Each th element represents a table cell.

Example

```
<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

Try it Yourself »

Vertical Table Headers

To use the first column as table headers, define the first cell in each row as a `<th>` element:

Example

```
<table>
  <tr>
    <th>Firstname</th>
    <td>Jill</td>
    <td>Eve</td>
  </tr>
  <tr>
    <th>Lastname</th>
    <td>Smith</td>
    <td>Jackson</td>
  </tr>
  <tr>
    <th>Age</th>
    <td>94</td>
    <td>50</td>
  </tr>
</table>
```

Try it Yourself »

Align Table Headers

By default, table headers are bold and centered:

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94

To left-align the table headers, use the CSS `text-align` property:

Example

```
th {
  text-align: left;
}
```

Try it Yourself »

Header for Multiple Columns

You can have a header that spans over two or more columns.

Name		Age
Jill	Smith	50
Eve	Jackson	94

To do this, use the `colspan` attribute on the `<th>` element:

Example

```
<table>
  <tr>
    <th colspan="2">Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

Try it Yourself »

You will learn more about colspan and rowspan in the Table colspan & rowspan chapter.

Table Caption

You can add a caption that serves as a heading for the entire table.

Monthly savings

Month	Savings
January	$100
February	$50

To add a caption to a table, use the `<caption>` tag:

Example

```
<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>
```

Try it Yourself »

Note: The `<caption>` tag should be inserted immediately after the `<table>` tag.

HTML Exercises

Exercise:

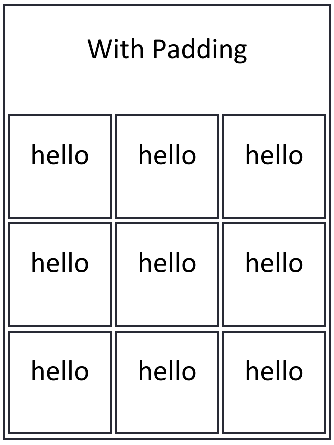
Add a table caption that says "Names".

```
<table>
  

  <tr>
    <th>First Name</th>
    <th>Last Name</th>
    <th>Points</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
</table>
```

Start the Exercise

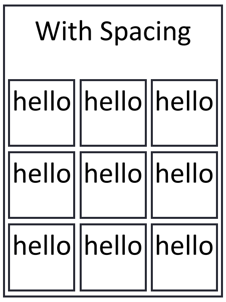
HTML Table Padding & Spacing

HTML tables can adjust the padding inside the cells, and also the space between the cells.



With Padding

hello	hello	hello
hello	hello	hello
hello	hello	hello



With Spacing

hello	hello	hello
hello	hello	hello
hello	hello	hello

HTML Table - Cell Padding

Cell padding is the space between the cell edges and the cell content.

By default the padding is set to 0.

To add padding on table cells, use the CSS `padding` property:

Example

```
th, td {
  padding: 15px;
}
```

Try it Yourself »

To add padding only above the content, use the `padding-top` property.

And the others sides with the `padding-bottom`, `padding-left`, and `padding-right` properties:

Example

```
th, td {
  padding-top: 10px;
  padding-bottom: 20px;
  padding-left: 30px;
  padding-right: 40px;
}
```

Try it Yourself »

HTML Table - Cell Spacing

Cell spacing is the space between each cell.

By default the space is set to 2 pixels.

To change the space between table cells, use the CSS `border-spacing` property on the `table` element:

Example

```
table {
  border-spacing: 30px;
}
```

Try it Yourself »

HTML Exercises

Exercise:

Use the correct CSS property to add 15 pixels of space between the cell border and cell content.

The result should look like this:

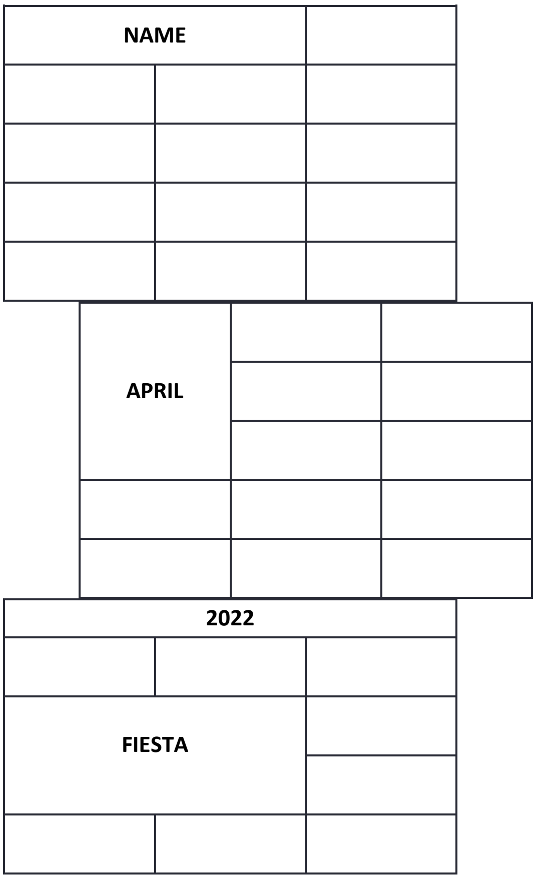
hello	hello	hello
hello	hello	hello
hello	hello	hello

```
.table td {
  ________: 15px;
}
```

Start the Exercise

HTML Table Colspan & Rowspan

HTML tables can have cells that span over multiple rows and/or columns.



HTML Table - Colspan

To make a cell span over multiple columns, use the colspan attribute:

Example

```
<table>
  <tr>
    <th colspan="2">Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>43</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>57</td>
  </tr>
</table>
```

Try it Yourself »

Note: The value of the colspan attribute represents the number of columns to span.

HTML Table - Rowspan

To make a cell span over multiple rows, use the rowspan attribute:

Example

```
<table>
  <tr>
    <th>Name</th>
    <td>Jill</td>
  </tr>
  <tr>
    <th rowspan="2">Phone</th>
    <td>555-1234</td>
  </tr>
  <tr>
    <td>555-8745</td>
</tr>
</table>
```

Try it Yourself »

Note: The value of the `rowspan` attribute represents the number of rows to span.

HTML Exercises

Exercise:

Use the correct HTML attribute to make the first TH element span two columns.

```
<table>
  <tr>
    <th [          ] >Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

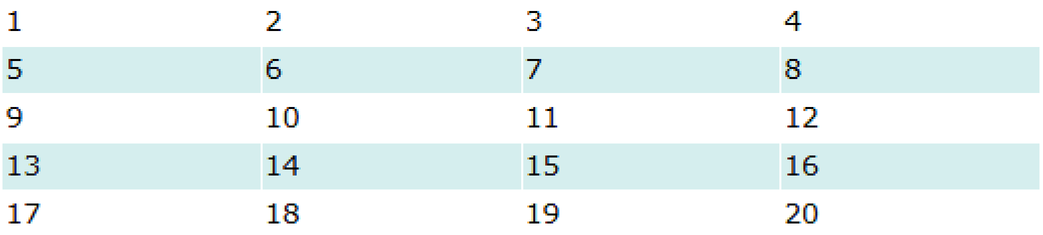
[Start the Exercise]

HTML Table Styling

Use CSS to make your tables look better.

HTML Table - Zebra Stripes

If you add a background colour on every other table row, you will get a nice zebra stripes effect.



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20

To style every other table row element, use the `:nth-child(even)` selector like this:

Example

```
tr:nth-child(even) {
  background-color: #D6EEEE;
}
```

Try it Yourself »

Note: If you use `(odd)` instead of `(even)`, the styling will occur on row 1,3,5 etc. instead of 2,4,6 etc.

HTML Table - Vertical Zebra Stripes

To make vertical zebra stripes, style every other *column*, instead of every other *row*.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20

Set the `:nth-child(even)` for table data elements like this:

Example

```
td:nth-child(even), th:nth-child(even) {
  background-color: #D6EEEE;
}
```

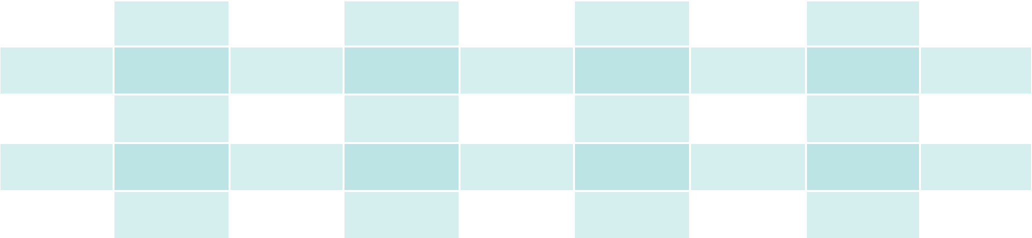
Try it Yourself »

Note: Put the `:nth-child()` selector on both `th` and `td` elements if you want to have the styling on both headers and regular table cells.

Combine Vertical and Horizontal Zebra Stripes

You can combine the styling from the two examples above and you will have stripes on every other row and every other column.

If you use a transparent colour you will get an overlapping effect.



Use an `rgba()` colour to specify the transparency of the colour:

Example

```
tr:nth-child(even) {
  background-color: rgba(150, 212, 212, 0.4);
}

th:nth-child(even),td:nth-child(even) {
  background-color: rgba(150, 212, 212, 0.4);
}
```

Try it Yourself »

Horizontal Dividers

First Name	Last Name	Savings
Peter	Griffin	$100
Lois	Griffin	$150
Joe	Swanson	$300

If you specify borders only at the bottom of each table row, you will have a table with horizontal dividers.

Add the `border-bottom` property to all `tr` elements to get horizontal dividers:

Example

```
tr {
  border-bottom: 1px solid #ddd;
}
```

Try it Yourself »

Hoverable Table

Use the `:hover` selector on `tr` to highlight table rows on mouse over:

First Name	Last Name	Savings
Peter	Griffin	$100
Lois	Griffin	$150
Joe	Swanson	$300

Example

```
tr:hover {background-color: #D6EEEE;}
```

Try it Yourself »

HTML Table Colgroup

The `<colgroup>` element is used to style specific columns of a table.

HTML Table Colgroup

If you want to style the two first columns of a table, use the `<colgroup>` and `<col>` elements.

MON	TUE	WED	THU	FRI	SAT	SUN
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

The `<colgroup>` element should be used as a container for the column specifications.

Each group is specified with a `<col>` element.

The `span` attribute specifies how many columns that get the style.

The `style` attribute specifies the style to give the columns.

Note: There is a very limited selection of legal CSS properties for colgroups.

Example

```
<table>
  <colgroup>
    <col span="2" style="background-color: #D6EEEE">
  </colgroup>
  <tr>
    <th>MON</th>
    <th>TUE</th>
    <th>WED</th>
    <th>THU</th>
...
```

Try it Yourself »

Note: The `<colgroup>` tag must be a child of a `<table>` element and should be placed before any other table elements, like `<thead>`, `<tr>`, `<td>` etc., but after the `<caption>` element, if present.

Legal CSS Properties

There is only a very limited selection of CSS properties that are allowed to be used in the colgroup:

`width` property
`visibility` property
`background` properties
`border` properties

All other CSS properties will have no effect on your tables.

Multiple Col Elements

If you want to style more columns with different styles, use more `<col>` elements inside the `<colgroup>`:

Example

```
<table>
  <colgroup>
    <col span="2" style="background-color: #D6EEEE">
    <col span="3" style="background-color: pink">
  </colgroup>
  <tr>
    <th>MON</th>
    <th>TUE</th>
    <th>WED</th>
    <th>THU</th>
...
```

Try it Yourself »

Empty Colgroups

If you want to style columns in the middle of a table, insert a "empty" `<col>` element (with no styles) for the columns before:

Example

```
<table>
  <colgroup>
    <col span="3">
    <col span="2" style="background-color: pink">
  </colgroup>
  <tr>
    <th>MON</th>
    <th>TUE</th>
    <th>WED</th>
    <th>THU</th>
...
```

Try it Yourself »

Hide Columns

You can hide columns with the `visibility: collapse` property:

Example

```
<table>
  <colgroup>
    <col span="2">
    <col span="3" style="visibility: collapse">
  </colgroup>
  <tr>
    <th>MON</th>
    <th>TUE</th>
    <th>WED</th>
    <th>THU</th>
...
```

Try it Yourself »

HTML Lists

HTML lists allow web developers to group a set of related items in lists.

Example

An unordered HTML list:

- Item
- Item
- Item
- Item

An ordered HTML list:

1. First item
2. Second item
3. Third item
4. Fourth item

Try it Yourself »

Unordered HTML List

An unordered list starts with the `<ul>` tag. Each list item starts with the `<li>` tag.

The list items will be marked with bullets (small black circles) by default:

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Try it Yourself »

Ordered HTML List

An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag.

The list items will be marked with numbers by default:

Example

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Try it Yourself »

HTML Description Lists

HTML also supports description lists.

A description list is a list of terms, with a description of each term.

The `<dl>` tag defines the description list, the `<dt>` tag defines the term (name), and the `<dd>` tag describes each term:

Example

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Try it Yourself »

HTML Exercises

Exercise:

Add a list item with the text "Coffee" inside the <ul> element.

```
<ul>□Coffee□</ul>
```

Start the Exercise

HTML List Tags

Tag	Description
<ul>	Defines an unordered list
<ol>	Defines an ordered list
<li>	Defines a list item
<dl>	Defines a description list
<dt>	Defines a term in a description list
<dd>	Describes the term in a description list

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Unordered Lists

The HTML <ul> tag defines an unordered (bulleted) list.

Unordered HTML List

An unordered list starts with the `<ul>` tag. Each list item starts with the `<li>` tag.

The list items will be marked with bullets (small black circles) by default:

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Try it Yourself »

Unordered HTML List - Choose List Item Marker

The CSS `list-style-type` property is used to define the style of the list item marker. It can have one of the following values:

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

Example - Disc

```
<ul style="list-style-type:disc;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Try it Yourself »

Example - Circle

```
<ul style="list-style-type:circle;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Try it Yourself »

Example - Square

```
<ul style="list-style-type:square;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Try it Yourself »

Example - None

```
<ul style="list-style-type:none;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Try it Yourself »

Nested HTML Lists

Lists can be nested (list inside list):

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

Try it Yourself »

Note: A list item (`<li>`) can contain a new list, and other HTML elements, like images and links, etc.

Horizontal List with CSS

HTML lists can be styled in many different ways with CSS.

One popular way is to style a list horizontally, to create a navigation menu:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333333;
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 16px;
  text-decoration: none;
}

li a:hover {
  background-color: #111111;
}
</style>
</head>
<body>

<ul>
```

```
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

</body>
</html>
```

Try it Yourself »

Tip: You can learn much more about CSS in our CSS Tutorial.

Chapter Summary

- Use the HTML `<ul>` element to define an unordered list
- Use the CSS `list-style-type` property to define the list item marker
- Use the HTML `<li>` element to define a list item
- Lists can be nested
- List items can contain other HTML elements
- Use the CSS property `float:left` to display a list horizontally

HTML Exercises

Exercise:

Finish the HTML code to make an unordered list.

```

  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>

```

Start the Exercise

HTML List Tags

Tag	Description
<ul>	Defines an unordered list

<ol>	Defines an ordered list
<li>	Defines a list item
<dl>	Defines a description list
<dt>	Defines a term in a description list
<dd>	Describes the term in a description list

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Ordered Lists

The HTML `<ol>` tag defines an ordered list. An ordered list can be numerical or alphabetical.

Ordered HTML List

An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag.

The list items will be marked with numbers by default:

Example

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Try it Yourself »

Ordered HTML List - The Type Attribute

The `type` attribute of the `<ol>` tag, defines the type of the list item marker:

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

Numbers:

```
<ol type="1">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Try it Yourself »

Uppercase Letters:

```
<ol type="A">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Try it Yourself »

Lowercase Letters:

```
<ol type="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Try it Yourself »

Uppercase Roman Numbers:

```
<ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Try it Yourself »

Lowercase Roman Numbers:

```
<ol type="i">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Try it Yourself »

Control List Counting

By default, an ordered list will start counting from 1. If you want to start counting from a specified number, you can use the `start` attribute:

Example

```
<ol start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Try it Yourself »

Nested HTML Lists

Lists can be nested (list inside list):

Example

```
<ol>
  <li>Coffee</li>
  <li>Tea
    <ol>
      <li>Black tea</li>
      <li>Green tea</li>
    </ol>
  </li>
  <li>Milk</li>
</ol>
```

Try it Yourself »

Note: A list item (`<li>`) can contain a new list, and other HTML elements, like images and links, etc.

Chapter Summary

- Use the HTML `<ol>` element to define an ordered list
- Use the HTML `type` attribute to define the numbering type
- Use the HTML `<li>` element to define a list item
- Lists can be nested
- List items can contain other HTML elements

HTML Exercises

Exercise:

Finish the HTML code to make an ordered list.

```
□
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
□
```

Start the Exercise

HTML List Tags

Tag	Description

<ul>	Defines an unordered list
<ol>	Defines an ordered list
<li>	Defines a list item
<dl>	Defines a description list
<dt>	Defines a term in a description list
<dd>	Describes the term in a description list

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Other Lists

HTML also supports description lists.

HTML Description Lists

A description list is a list of terms, with a description of each term.

The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term:

Example

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Try it Yourself »

Chapter Summary

- Use the HTML `<dl>` element to define a description list
- Use the HTML `<dt>` element to define the description term
- Use the HTML `<dd>` element to describe the term in a description list

HTML Exercises

Exercise:

Use the correct HTML elements to make a description of each term in the description list.

```
<dl>
  <dt>Coffee</dt>
  □ - black hot drink□
  <dt>Milk</dt>
  □ - white cold drink□
</dl>
```

Start the Exercise

HTML List Tags

Tag	Description
<ul>	Defines an unordered list
<ol>	Defines an ordered list
<li>	Defines a list item
<dl>	Defines a description list

<dt>	Defines a term in a description list
<dd>	Describes the term in a description list

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Block and Inline Elements

Every HTML element has a default display value, depending on what type of element it is.

The two most common display values are block and inline.

Block-level Elements

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.

A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

Two commonly used block elements are: `<p>` and `<div>`.

The `<p>` element defines a paragraph in an HTML document.

The `<div>` element defines a division or a section in an HTML document.

The <p> element is a block-level element.

The <div> element is a block-level element.

Example

```
<p>Hello World</p>
<div>Hello World</div>
```

Try it Yourself »

Here are the block-level elements in HTML:

```
<address>
```

```
<article>

<aside>

<blockquote>

<canvas>

<dd>

<div>

<dl>

<dt>

<fieldset>

<figcaption>

<figure>

<footer>

<form>

<h1>-<h6>

<header>

<hr>

<li>

<main>

<nav>

<noscript>

<ol>

<p>

<pre>

<section>

<table>

<tfoot>
```

<ul>

<video>

Inline Elements

An inline element does not start on a new line.

An inline element only takes up as much width as necessary.

This is a <span> element inside a paragraph.

Example

```
<span>Hello World</span>
```

Try it Yourself »

Here are the inline elements in HTML:

<a>

<abbr>

<acronym>

<b>

<bdo>

<big>

<br>

<button>

<cite>

<code>

<dfn>

<em>

<i>

<img>

<input>

<kbd>

<label>

<map>

<object>

<output>

<q>

<samp>

<script>

<select>

<small>

<span>

<strong>

<sub>

<sup>

<textarea>

<time>

<tt>

<var>

Note: An inline element cannot contain a block-level element!

The <div> Element

The <div> element is often used as a container for other HTML elements.

The <div> element has no required attributes, but style, class and id are common.

When used together with CSS, the <div> element can be used to style blocks of content:

Example

```
<div style="background-color:black;color:white;padding:20px;">
  <h2>London</h2>
  <p>London is the capital city of England. It is the most populous
city in the United Kingdom, with a metropolitan area of over 13 million
inhabitants.</p>
</div>
```

Try it Yourself »

You will learn more about the <div> element in the next chapter.

The <span> Element

The <span> element is an inline container used to mark up a part of a text, or a part of a document.

The <span> element has no required attributes, but style, class and id are common.

When used together with CSS, the <span> element can be used to style parts of the text:

Example

```
<p>My mother has <span style="color:blue;font-
weight:bold;">blue</span> eyes and my father
has <span style="color:darkolivegreen;font-weight:bold;">dark
green</span> eyes.</p>
```

Try it Yourself »

Chapter Summary

- A block-level element always starts on a new line and takes up the full width available
- An inline element does not start on a new line and it only takes up as much width as necessary
- The <div> element is a block-level and is often used as a container for other HTML elements
- The <span> element is an inline container used to mark up a part of a text, or a part of a document

HTML Exercises

Exercise:

Name one HTML block element.

Start the Exercise

HTML Tags

Tag	Description
<div>	Defines a section in a document (block-level)
<span>	Defines a section in a document (inline)

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Div Element

The <div> element is used as a container for other HTML elements.

The <div> Element

The <div> element is by default a block element, meaning that it takes all available width, and comes with line breaks before and after.

Example

A <div> element takes up all available width:

```
Lorem Ipsum <div>I am a div</div> dolor sit amet.
```

Result

Lorem Ipsum

I am a div

dolor sit amet.

Try it Yourself »

The `<div>` element has no required attributes, but `style`, `class` and `id` are common.

<div> as a container

The `<div>` element is often used to group sections of a web page together.

Example

A <div> element with HTML elements:

```
<div>
  <h2>London</h2>
  <p>London is the capital city of England.</p>
  <p>London has over 13 million inhabitants.</p>
</div>
```

Result

London

London is the capital city of England.

London has over 13 million inhabitants.

Try it Yourself »

Center align a <div> element

If you have a `<div>` element that is not 100% wide, and you want to center-align it, set the CSS `margin` property to `auto`.

Example

```
<style>
div {
  width:300px;
  margin:auto;
}
</style>
```

Result

London

London is the capital city of England.

London has over 13 million inhabitants.

Try it Yourself »

Multiple <div> elements

You can have many `<div>` containers on the same page.

Example

```
<div>
  <h2>London</h2>
  <p>London is the capital city of England.</p>
  <p>London has over 13 million inhabitants.</p>
</div>

<div>
  <h2>Oslo</h2>
  <p>Oslo is the capital city of Norway.</p>
  <p>Oslo has over 600.000 inhabitants.</p>
</div>

<div>
  <h2>Rome</h2>
  <p>Rome is the capital city of Italy.</p>
  <p>Rome has almost 3 million inhabitants.</p>
</div>
```

Result

London

London is the capital city of England.

London has over 13 million inhabitants.

Oslo

Oslo is the capital city of Norway.

Oslo has over 600.000 inhabitants.

Rome

Rome is the capital city of Italy.

Rome has almost 3 million inhabitants.

Try it Yourself »

Aligning <div> elements side by side

When building web pages, you often want to have two or more `<div>` elements side by side, like this:

London

London is the capital city of England.

London has over 13 million inhabitants.

Oslo

Oslo is the capital city of Norway.

Oslo has over 600.000 inhabitants.

Rome

Rome is the capital city of Italy.

Rome has almost 3 million inhabitants.

There are different methods for aligning elements side by side, all include some CSS styling. We will look at the most common methods:

Float

The CSS `float` property was not originally meant to align `<div>` elements side-by-side, but has been used for this purpose for many years.

The CSS `float` property is used for positioning and formatting content and allow elements float next to each other instead of on top of each other.

Example

How to use float to align div elements side by side:

```
<style>
.mycontainer {
  width:100%;
  overflow:auto;
}
.mycontainer div {
  width:33%;
  float:left;
}
</style>
```

Result

London

London is the capital city of England.

London has over 13 million inhabitants.

Oslo

Oslo is the capital city of Norway.

Oslo has over 600.000 inhabitants.

Rome

Rome is the capital city of Italy.

Rome has almost 3 million inhabitants.

Try it Yourself »

Learn more about float in our CSS float tutorial.

Inline-block

If you change the `<div>` element's `display` property from `block` to `inline-block`, the `<div>` elements will no longer add a line break before and after, and will be displayed side by side instead of on top of each other.

Example

How to use display: inline-block to align div elements side by side:

```
<style>
div {
  width: 30%;
  display: inline-block;
}
</style>
```

Result

London

London is the capital city of England.

London has over 13 million inhabitants.

Oslo

Oslo is the capital city of Norway.

Oslo has over 600.000 inhabitants.

Rome

Rome is the capital city of Italy.

Rome has almost 3 million inhabitants.

Try it Yourself »

Flex

The CSS Flexbox Layout Module was introduced to make it easier to design flexible responsive layout structure without using float or positioning.

To make the CSS flex method work, surround the `<div>` elements with another `<div>` element and give it the status as a flex container.

Example

How to use flex to align div elements side by side:

```
<style>
.mycontainer {
  display: flex;
}
.mycontainer > div {
  width:33%;
}
</style>
```

Result

London

London is the capital city of England.

London has over 13 million inhabitants.

Oslo

Oslo is the capital city of Norway.

Oslo has over 600.000 inhabitants.

Rome

Rome is the capital city of Italy.

Rome has almost 3 million inhabitants.

Try it Yourself »

Learn more about flex in our CSS flexbox tutorial.

Grid

The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.

Sounds almost the same as flex, but has the ability to define more than one row and position each row individually.

The CSS grid method requires that you surround the `<div>` elements with another `<div>` element and give the status as a grid container, and you must specify the width of each column.

Example

How to use grid to align <div> elements side by side:

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: 33% 33% 33%;
}
</style>
```

Result

London

London is the capital city of England.

London has over 13 million inhabitants.

Oslo

Oslo is the capital city of Norway.

Oslo has over 600.000 inhabitants.

Rome

Rome is the capital city of Italy.

Rome has almost 3 million inhabitants.

Try it Yourself »

Learn more about grid in our CSS grid tutorial.

HTML Tags

Tag	Description

<div>	Defines a section in a document (block-level)

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML class Attribute

The HTML `class` attribute is used to specify a class for an HTML element.

Multiple HTML elements can share the same class.

Using The class Attribute

The `class` attribute is often used to point to a class name in a style sheet. It can also be used by a JavaScript to access and manipulate elements with the specific class name.

In the following example we have three `<div>` elements with a `class` attribute with the value of "city". All of the three `<div>` elements will be styled equally according to the `.city` style definition in the head section:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  border: 2px solid black;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<div class="city">
  <h2>London</h2>
  <p>London is the capital of England.</p>
</div>

<div class="city">
  <h2>Paris</h2>
  <p>Paris is the capital of France.</p>
</div>

<div class="city">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan.</p>
</div>

</body>
</html>
```

Try it Yourself »

In the following example we have two `<span>` elements with a `class` attribute with the value of "note". Both `<span>` elements will be styled equally according to the `.note` style definition in the head section:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
.note {
  font-size: 120%;
  color: red;
}
</style>
</head>
<body>

<h1>My <span class="note">Important</span> Heading</h1>
<p>This is some <span class="note">important</span> text.</p>

</body>
</html>
```

Try it Yourself »

Tip: The `class` attribute can be used on **any** HTML element.

Note: The class name is case sensitive!

Tip: You can learn much more about CSS in our CSS Tutorial.

The Syntax For Class

To create a class; write a period (.) character, followed by a class name. Then, define the CSS properties within curly braces {}:

Example

Create a class named "city":

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>
</head>
<body>

<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>

</body>
</html>
```

Try it Yourself »

Multiple Classes

HTML elements can belong to more than one class.

To define multiple classes, separate the class names with a space, e.g. <div class="city main">. The element will be styled according to all the classes specified.

In the following example, the first `<h2>` element belongs to both the `city` class and also to the `main` class, and will get the CSS styles from both of the classes:

Example

```
<h2 class="city main">London</h2>
<h2 class="city">Paris</h2>
<h2 class="city">Tokyo</h2>
```

Try it Yourself »

Different Elements Can Share Same Class

Different HTML elements can point to the same class name.

In the following example, both `<h2>` and `<p>` point to the "city" class and will share the same style:

Example

```
<h2 class="city">Paris</h2>
<p class="city">Paris is the capital of France</p>
```

Try it Yourself »

Use of The class Attribute in JavaScript

The class name can also be used by JavaScript to perform certain tasks for specific elements.

JavaScript can access elements with a specific class name with the `getElementsByClassName()` method:

Example

Click on a button to hide all elements with the class name "city":

```
<script>
function myFunction() {
  var x = document.getElementsByClassName("city");
  for (var i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
}
</script>
```

Try it Yourself »

Don't worry if you don't understand the code in the example above.

You will learn more about JavaScript in our HTML JavaScript chapter, or you can study our JavaScript Tutorial.

Chapter Summary

- The HTML `class` attribute specifies one or more class names for an element
- Classes are used by CSS and JavaScript to select and access specific elements
- The `class` attribute can be used on any HTML element
- The class name is case sensitive
- Different HTML elements can point to the same class name
- JavaScript can access elements with a specific class name with the `getElementsByClassName()` method

HTML Exercises

Exercise:

Create a class selector named "special".

Add a color property with the value "blue" inside the "special" class.

```
<!DOCTYPE html>
<html>
<head>
<style>
□ □
  □;
□
</style>
</head>
<body>

<p class="special">My paragraph</p>

</body>
</html>
```

Start the Exercise

HTML id Attribute

The HTML `id` attribute is used to specify a unique id for an HTML element.

You cannot have more than one element with the same id in an HTML document.

Using The id Attribute

The id attribute specifies a unique id for an HTML element. The value of the id attribute must be unique within the HTML document.

The id attribute is used to point to a specific style declaration in a style sheet. It is also used by JavaScript to access and manipulate the element with the specific id.

The syntax for id is: write a hash character (#), followed by an id name. Then, define the CSS properties within curly braces {}.

In the following example we have an `<h1>` element that points to the id name "myHeader". This `<h1>` element will be styled according to the `#myHeader` style definition in the head section:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>
</head>
<body>

<h1 id="myHeader">My Header</h1>

</body>
</html>
```

Try it Yourself »

Note: The id name is case sensitive!

Note: The id name must contain at least one character, cannot start with a number, and must not contain whitespaces (spaces, tabs, etc.).

Difference Between Class and ID

A class name can be used by multiple HTML elements, while an id name must only be used by one HTML element within the page:

Example

```
<style>
/* Style the element with the id "myHeader" */
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}

/* Style all elements with the class name "city" */
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>

<!-- An element with a unique id -->
<h1 id="myHeader">My Cities</h1>

<!-- Multiple elements with same class -->
<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>
```

Try it Yourself »

Tip: You can learn much more about CSS in our CSS Tutorial.

HTML Bookmarks with ID and Links

HTML bookmarks are used to allow readers to jump to specific parts of a webpage.

Bookmarks can be useful if your page is very long.

To use a bookmark, you must first create it, and then add a link to it.

Then, when the link is clicked, the page will scroll to the location with the bookmark.

Example

First, create a bookmark with the id attribute:

```
<h2 id="C4">Chapter 4</h2>
```

Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:

Example

```
<a href="#C4">Jump to Chapter 4</a>
```

Try it Yourself »

Or, add a link to the bookmark ("Jump to Chapter 4"), from another page:

```
<a href="html_demo.html#C4">Jump to Chapter 4</a>
```

Using The id Attribute in JavaScript

The id attribute can also be used by JavaScript to perform some tasks for that specific element.

JavaScript can access an element with a specific id with the getElementById() method:

Example

Use the id attribute to manipulate text with JavaScript:

```
<script>
function displayResult() {
  document.getElementById("myHeader").innerHTML = "Have a nice day!";
}
</script>
```

Try it Yourself »

Tip: Study JavaScript in the HTML JavaScript chapter, or in our JavaScript Tutorial.

Chapter Summary

- The id attribute is used to specify a unique id for an HTML element
- The value of the id attribute must be unique within the HTML document

- The id attribute is used by CSS and JavaScript to style/select a specific element
- The value of the id attribute is case sensitive
- The id attribute is also used to create HTML bookmarks
- JavaScript can access an element with a specific id with the getElementById() method

HTML Exercises

Exercise:

Add the correct HTML attribute to make the H1 element red.

```
<!DOCTYPE html>
<html>
<head>
<style>
#myheader {color:red;}
</style>
</head>
<body>

<h1 □ >My Home Page</h1>

</body>
</html>
```

Start the Exercise

HTML Iframes

An HTML iframe is used to display a web page within a web page.

HTML Iframe Syntax

The HTML <iframe> tag specifies an inline frame.

An inline frame is used to embed another document within the current HTML document.

Syntax

```
<iframe src="url" title="description"></iframe>
```

Tip: It is a good practice to always include a `title` attribute for the `<iframe>`. This is used by screen readers to read out what the content of the iframe is.

Iframe - Set Height and Width

Use the `height` and `width` attributes to specify the size of the iframe.

The height and width are specified in pixels by default:

Example

```
<iframe src="demo_iframe.htm" height="200" width="300" title="Iframe
Example"></iframe>
```

Try it Yourself »

Or you can add the `style` attribute and use the CSS `height` and `width` properties:

Example

```
<iframe src="demo_iframe.htm" style="height:200px;width:300px;" title="
Iframe Example"></iframe>
```

Try it Yourself »

Iframe - Remove the Border

By default, an iframe has a border around it.

To remove the border, add the `style` attribute and use the CSS `border` property:

Example

```
<iframe src="demo_iframe.htm" style="border:none;" title="Iframe
Example"></iframe>
```

Try it Yourself »

With CSS, you can also change the size, style and color of the iframe's border:

Example

```
<iframe src="demo_iframe.htm" style="border:2px solid
red;" title="Iframe Example"></iframe>
```

Try it Yourself »

Iframe - Target for a Link

An iframe can be used as the target frame for a link.

The `target` attribute of the link must refer to the `name` attribute of the iframe:

Example

```
<iframe src="demo_iframe.htm" name="iframe_a" title="Iframe
Example"></iframe>

<p><a href="https://www.w3schools.com" target="iframe_a">W3Schools.com<
/a></p>
```

Try it Yourself »

Chapter Summary

- The HTML `<iframe>` tag specifies an inline frame
- The `src` attribute defines the URL of the page to embed
- Always include a `title` attribute (for screen readers)
- The `height` and `width` attributes specify the size of the iframe
- Use `border:none;` to remove the border around the iframe

HTML Exercises

Exercise:

Create an iframe with a URL address that goes to https://www.w3schools.com.

```
<iframe □="https://www.w3schools.com"></iframe>
```

Start the Exercise

HTML iframe Tag

Tag	Description
<iframe>	Defines an inline frame

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML JavaScript

JavaScript makes HTML pages more dynamic and interactive.

Example

My First JavaScript

Click me to display Date and Time

Try it Yourself »

The HTML <script> Tag

The HTML `<script>` tag is used to define a client-side script (JavaScript).

The `<script>` element either contains script statements, or it points to an external script file through the `src` attribute.

Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

To select an HTML element, JavaScript most often uses the `document.getElementById()` method.

This JavaScript example writes "Hello JavaScript!" into an HTML element with id="demo":

Example

```
<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
```

Try it Yourself »

Tip: You can learn much more about JavaScript in our JavaScript Tutorial.

A Taste of JavaScript

Here are some examples of what JavaScript can do:

Example

JavaScript can change content:

```
document.getElementById("demo").innerHTML = "Hello JavaScript!";
```

Try it Yourself »

Example

JavaScript can change styles:

```
document.getElementById("demo").style.fontSize = "25px";
document.getElementById("demo").style.color = "red";
document.getElementById("demo").style.backgroundColor = "yellow";
```

Try it Yourself »

Example

JavaScript can change attributes:

```
document.getElementById("image").src = "picture.gif";
```

Try it Yourself »

The HTML <noscript> Tag

The HTML `<noscript>` tag defines an alternate content to be displayed to users that have disabled scripts in their browser or have a browser that doesn't support scripts:

Example

```
<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
<noscript>Sorry, your browser does not support JavaScript!</noscript>
```

Try it Yourself »

HTML Exercises

Exercise:

Use JavaScript to change the HTML content of the `<p>` element to "Hello World!".

```
<body>

<p id="demo">Hi.</p>

<script>
```

```
document.        ("demo").innerHTML = "Hello World!";
</script>

</body>
```

Start the Exercise

HTML Script Tags

Tag	Description
<script>	Defines a client-side script
<noscript>	Defines an alternate content for users that do not support client-side scripts

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML File Paths

A file path describes the location of a file in a web site's folder structure.

File Path Examples

Path	Description
<img src="picture.jpg">	The "picture.jpg" file is located in the same folder as the current page
<img src="images/picture.jpg">	The "picture.jpg" file is located in the images folder in the current folder

<img src="/images/picture.jpg">	The "picture.jpg" file is located in the images folder at the root of the current web
<img src="../picture.jpg">	The "picture.jpg" file is located in the folder one level up from the current folder

HTML File Paths

A file path describes the location of a file in a web site's folder structure.

File paths are used when linking to external files, like:

- Web pages
- Images
- Style sheets
- JavaScripts

Absolute File Paths

An absolute file path is the full URL to a file:

Example

```
<img src="https://www.w3schools.com/images/picture.jpg" alt="Mountain">
```

Try it Yourself »

The <img> tag is explained in the chapter: HTML Images.

Relative File Paths

A relative file path points to a file relative to the current page.

In the following example, the file path points to a file in the images folder located at the root of the current web:

Example

```
<img src="/images/picture.jpg" alt="Mountain">
```

Try it Yourself »

In the following example, the file path points to a file in the images folder located in the current folder:

Example

```
<img src="images/picture.jpg" alt="Mountain">
```

Try it Yourself »

In the following example, the file path points to a file in the images folder located in the folder one level up from the current folder:

Example

```
<img src="../images/picture.jpg" alt="Mountain">
```

Try it Yourself »

Best Practice

It is best practice to use relative file paths (if possible).

When using relative file paths, your web pages will not be bound to your current base URL. All links will work on your own computer (localhost) as well as on your current public domain and your future public domains.

HTML - The Head Element

The HTML `<head>` element is a container for the following elements: `<title>`, `<style>`, `<meta>`, `<link>`, `<script>`, and `<base>`.

The HTML <head> Element

The `<head>` element is a container for metadata (data about data) and is placed between the `<html>` tag and the `<body>` tag.

HTML metadata is data about the HTML document. Metadata is not displayed.

Metadata typically define the document title, character set, styles, scripts, and other meta information.

The HTML <title> Element

The `<title>` element defines the title of the document. The title must be text-only, and it is shown in the browser's title bar or in the page's tab.

The `<title>` element is required in HTML documents!

The content of a page title is very important for search engine optimization (SEO)! The page title is used by search engine algorithms to decide the order when listing pages in search results.

The `<title>` element:

- defines a title in the browser toolbar
- provides a title for the page when it is added to favorites
- displays a title for the page in search engine-results

So, try to make the title as accurate and meaningful as possible!

A simple HTML document:

Example

```
<!DOCTYPE html>
<html>
<head>
  <title>A Meaningful Page Title</title>
</head>
<body>

The content of the document......

</body>
</html>
```

Try it Yourself »

The HTML <style> Element

The `<style>` element is used to define style information for a single HTML page:

Example

```
<style>
  body {background-color: powderblue;}
  h1 {color: red;}
  p {color: blue;}
</style>
```

Try it Yourself »

The HTML <link> Element

The `<link>` element defines the relationship between the current document and an external resource.

The `<link>` tag is most often used to link to external style sheets:

Example

```
<link rel="stylesheet" href="mystyle.css">
```

Try it Yourself »

Tip: To learn all about CSS, visit our CSS Tutorial.

The HTML <meta> Element

The `<meta>` element is typically used to specify the character set, page description, keywords, author of the document, and viewport settings.

The metadata will not be displayed on the page, but is used by browsers (how to display content or reload page), by search engines (keywords), and other web services.

Examples

Define the character set used:

```
<meta charset="UTF-8">
```

Define keywords for search engines:

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

Define a description of your web page:

```
<meta name="description" content="Free Web tutorials">
```

Define the author of a page:

```
<meta name="author" content="John Doe">
```

Refresh document every 30 seconds:

```
<meta http-equiv="refresh" content="30">
```

Setting the viewport to make your website look good on all devices:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Example of `<meta>` tags:

Example

```
<meta charset="UTF-8">
<meta name="description" content="Free Web tutorials">
<meta name="keywords" content="HTML, CSS, JavaScript">
<meta name="author" content="John Doe">
```

Try it Yourself »

Setting The Viewport

The viewport is the user's visible area of a web page. It varies with the device - it will be smaller on a mobile phone than on a computer screen.

You should include the following `<meta>` element in all your web pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

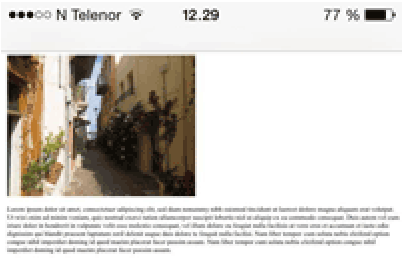
This gives the browser instructions on how to control the page's dimensions and scaling.

The `width=device-width` part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

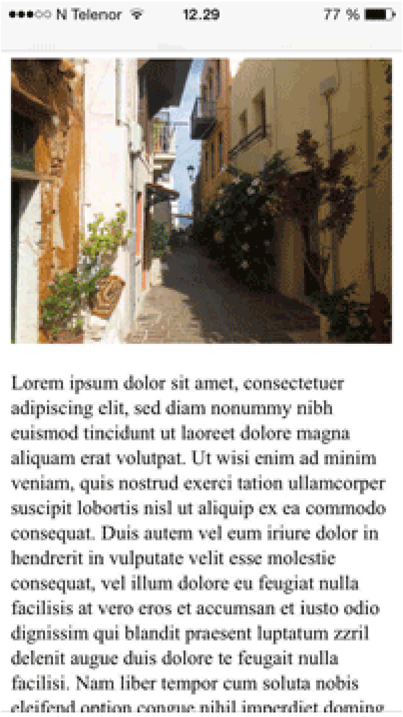
The `initial-scale=1.0` part sets the initial zoom level when the page is first loaded by the browser.

Here is an example of a web page *without* the viewport meta tag, and the same web page *with* the viewport meta tag:

Tip: If you are browsing this page with a phone or a tablet, you can click on the two links below to see the difference.



[Without the viewport meta tag]



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming

[With the viewport meta tag]

The HTML <script> Element

The `<script>` element is used to define client-side JavaScripts.

The following JavaScript writes "Hello JavaScript!" into an HTML element with id="demo":

Example

```
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Hello JavaScript!";
}
</script>
```

Try it Yourself »

Tip: To learn all about JavaScript, visit our JavaScript Tutorial.

The HTML <base> Element

The `<base>` element specifies the base URL and/or target for all relative URLs in a page.

The `<base>` tag must have either an href or a target attribute present, or both.

There can only be one single `<base>` element in a document!

Example

Specify a default URL and a default target for all links on a page:

```
<head>
<base href="https://www.w3schools.com/" target="_blank">
</head>

<body>
<img src="images/stickman.gif" width="24" height="39" alt="Stickman">
<a href="tags/tag_base.asp">HTML base Tag</a>
</body>
```

Try it Yourself »

Chapter Summary

- The `<head>` element is a container for metadata (data about data)
- The `<head>` element is placed between the `<html>` tag and the `<body>` tag
- The `<title>` element is required and it defines the title of the document
- The `<style>` element is used to define style information for a single document

- The `<link>` tag is most often used to link to external style sheets
- The `<meta>` element is typically used to specify the character set, page description, keywords, author of the document, and viewport settings
- The `<script>` element is used to define client-side JavaScripts
- The `<base>` element specifies the base URL and/or target for all relative URLs in a page

HTML head Elements

Tag	Description
<head>	Defines information about the document
<title>	Defines the title of a document
<base>	Defines a default address or a default target for all links on a page
<link>	Defines the relationship between a document and an external resource
<meta>	Defines metadata about an HTML document
<script>	Defines a client-side script
<style>	Defines style information for a document

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Layout Elements and Techniques

Websites often display content in multiple columns (like a magazine or a newspaper).

Example

Cities

- London
- Paris
- Tokyo

London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

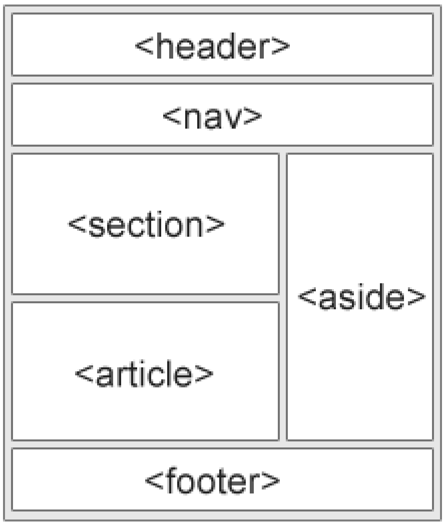
Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

Footer

Try it Yourself »

HTML Layout Elements

HTML has several semantic elements that define the different parts of a web page:



- <header> - Defines a header for a document or a section
- <nav> - Defines a set of navigation links
- <section> - Defines a section in a document
- <article> - Defines an independent, self-contained content
- <aside> - Defines content aside from the content (like a sidebar)
- <footer> - Defines a footer for a document or a section
- <details> - Defines additional details that the user can open and close on demand
- <summary> - Defines a heading for the <details> element

You can read more about semantic elements in our HTML Semantics chapter.

HTML Layout Techniques

There are four different techniques to create multicolumn layouts. Each technique has its pros and cons:

- CSS framework
- CSS float property
- CSS flexbox
- CSS grid

CSS Frameworks

If you want to create your layout fast, you can use a CSS framework, like W3.CSS or Bootstrap.

CSS Float Layout

It is common to do entire web layouts using the CSS `float` property. Float is easy to learn - you just need to remember how the `float` and `clear` properties work. **Disadvantages:** Floating elements are tied to the document flow, which may harm the flexibility. Learn more about float in our CSS Float and Clear chapter.

Example

Cities

- London
- Paris
- Tokyo

London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

Footer

Try it Yourself »

CSS Flexbox Layout

Use of flexbox ensures that elements behave predictably when the page layout must accommodate different screen sizes and different display devices.

Learn more about flexbox in our CSS Flexbox chapter.

Example

Cities

- London
- Paris
- Tokyo

London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

Footer

Try it Yourself »

CSS Grid Layout

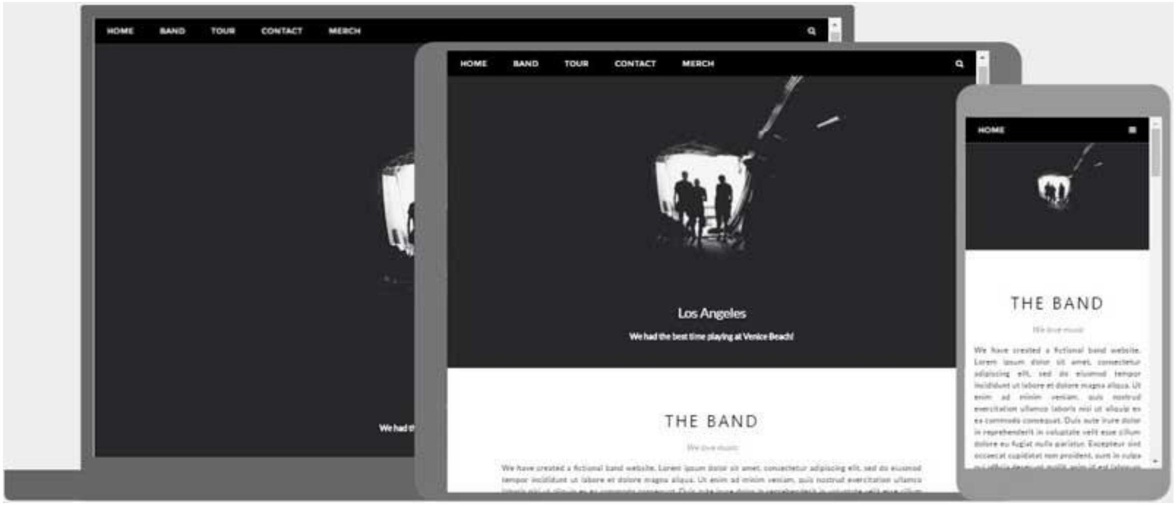
The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.

Learn more about CSS grids in our CSS Grid Intro chapter.

HTML Responsive Web Design

Responsive web design is about creating web pages that look good on all devices!

A responsive web design will automatically adjust for different screen sizes and viewports.



What is Responsive Web Design?

Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge, a website, to make it look good on all devices (desktops, tablets, and phones):

Try it Yourself »

Setting The Viewport

To create a responsive website, add the following `<meta>` tag to all your web pages:

Example

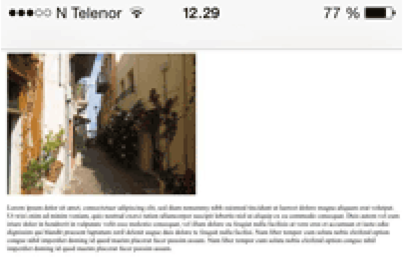
```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Try it Yourself »

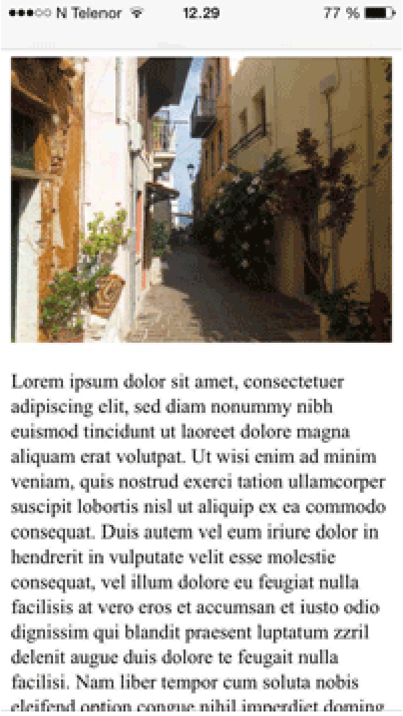
This will set the viewport of your page, which will give the browser instructions on how to control the page's dimensions and scaling.

Here is an example of a web page *without* the viewport meta tag, and the same web page *with* the viewport meta tag:

Without the viewport meta tag:



With the viewport meta tag:



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming

Tip: If you are browsing this page on a phone or a tablet, you can click on the two links above to see the difference.

Responsive Images

Responsive images are images that scale nicely to fit any browser size.

Using the width Property

If the CSS `width` property is set to 100%, the image will be responsive and scale up and down:

Example

```
<img src="img_girl.jpg" style="width:100%;">
```

Try it Yourself »

Notice that in the example above, the image can be scaled up to be larger than its original size. A better solution, in many cases, will be to use the `max-width` property instead.

Using the max-width Property

If the `max-width` property is set to 100%, the image will scale down if it has to, but never scale up to be larger than its original size:

Example

```
<img src="img_girl.jpg" style="max-width:100%;height:auto;">
```

Try it Yourself »

Show Different Images Depending on Browser Width

The HTML `<picture>` element allows you to define different images for different browser window sizes.

Resize the browser window to see how the image below changes depending on the width:

Example

```
<picture>
  <source srcset="img_smallflower.jpg" media="(max-width: 600px)">
  <source srcset="img_flowers.jpg" media="(max-width: 1500px)">
  <source srcset="flowers.jpg">
  <img src="img_smallflower.jpg" alt="Flowers">
</picture>
```

Try it Yourself »

Responsive Text Size

The text size can be set with a "vw" unit, which means the "viewport width".

That way the text size will follow the size of the browser window:

Hello World

Resize the browser window to see how the text size scales.

Example

```
<h1 style="font-size:10vw">Hello World</h1>
```

Try it Yourself »

Viewport is the browser window size. 1vw = 1% of viewport width. If the viewport is 50cm wide, 1vw is 0.5cm.

Media Queries

In addition to resize text and images, it is also common to use media queries in responsive web pages.

With media queries you can define completely different styles for different browser sizes.

Example: resize the browser window to see that the three div elements below will display horizontally on large screens and stack vertically on small screens:

Left Menu

Main Content

Right Content

Example

```
<style>
.left, .right {
  float: left;
  width: 20%; /* The width is 20%, by default */
}

.main {
  float: left;
  width: 60%; /* The width is 60%, by default */
}

/* Use a media query to add a breakpoint at 800px: */
@media screen and (max-width: 800px) {
  .left, .main, .right {
    width: 100%; /* The width is 100%, when the viewport is 800px or
smaller */
  }
}
</style>
```

Try it Yourself »

Tip: To learn more about Media Queries and Responsive Web Design, read our RWD Tutorial.

Responsive Web Page - Full Example

A responsive web page should look good on large desktop screens and on small mobile phones.

Try it Yourself »

Responsive Web Design - Frameworks

All popular CSS Frameworks offer responsive design.

They are free, and easy to use.

W3.CSS

W3.CSS is a modern CSS framework with support for desktop, tablet, and mobile design by default.

W3.CSS is smaller and faster than similar CSS frameworks.

W3.CSS is designed to be independent of jQuery or any other JavaScript library.

W3.CSS Demo

Resize the page to see the responsiveness!

London

London is the capital city of England.

It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Paris

Paris is the capital of France.

The Paris area is one of the largest population centers in Europe, with more than 12 million inhabitants.

Tokyo

Tokyo is the capital of Japan.

It is the center of the Greater Tokyo Area, and the most populous metropolitan area in the world.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>W3.CSS</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
</head>
<body>

<div class="w3-container w3-green">
  <h1>W3Schools Demo</h1>
  <p>Resize this responsive page!</p>
</div>

<div class="w3-row-padding">
  <div class="w3-third">
    <h2>London</h2>
    <p>London is the capital city of England.</p>
    <p>It is the most populous city in the United Kingdom,
    with a metropolitan area of over 13 million inhabitants.</p>
  </div>

  <div class="w3-third">
    <h2>Paris</h2>
    <p>Paris is the capital of France.</p>
    <p>The Paris area is one of the largest population centers in
Europe,
    with more than 12 million inhabitants.</p>
  </div>

  <div class="w3-third">
    <h2>Tokyo</h2>
    <p>Tokyo is the capital of Japan.</p>
    <p>It is the center of the Greater Tokyo Area,
    and the most populous metropolitan area in the world.</p>
  </div>
</div>

</body>
</html>
```

Try it Yourself »

To learn more about W3.CSS, read our W3.CSS Tutorial.

Bootstrap

Another popular CSS framework is Bootstrap:

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Bootstrap 5 Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>

<div class="container-fluid p-5 bg-primary text-white text-center">
  <h1>My First Bootstrap Page</h1>
  <p>Resize this responsive page to see the effect!</p>
</div>

<div class="container mt-5">
  <div class="row">
    <div class="col-sm-4">
      <h3>Column 1</h3>
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-sm-4">
      <h3>Column 2</h3>
      <p>Lorem ipsum...</p>
    </div>
    <div class="col-sm-4">
      <h3>Column 3</h3>
      <p>Lorem ipsum...</p>
    </div>
  </div>
</div>
```

Try it Yourself »

To learn more about Bootstrap, go to our Bootstrap Tutorial.

HTML Computer Code Elements

HTML contains several elements for defining user input and computer code.

Example

```
<code>
x = 5;
y = 6;
z = x + y;
</code>
```

Try it Yourself »

HTML <kbd> For Keyboard Input

The HTML `<kbd>` element is used to define keyboard input. The content inside is displayed in the browser's default monospace font.

Example

Define some text as keyboard input in a document:

```
<p>Save the document by pressing <kbd>Ctrl + S</kbd></p>
```

Result:

Save the document by pressing `Ctrl + S`

Try it Yourself »

HTML <samp> For Program Output

The HTML `<samp>` element is used to define sample output from a computer program. The content inside is displayed in the browser's default monospace font.

Example

Define some text as sample output from a computer program in a document:

```
<p>Message from my computer:</p>
<p><samp>File not found.<br>Press F1 to continue</samp></p>
```

Result:

Message from my computer:

```
File not found.
Press F1 to continue
```

Try it Yourself »

HTML <code> For Computer Code

The HTML `<code>` element is used to define a piece of computer code. The content inside is displayed in the browser's default monospace font.

Example

Define some text as computer code in a document:

```
<code>
x = 5;
y = 6;
z = x + y;
</code>
```

Result:

```
x = 5; y = 6; z = x + y;
```

Try it Yourself »

Notice that the `<code>` element does not preserve extra whitespace and line-breaks.

To fix this, you can put the `<code>` element inside a `<pre>` element:

Example

```
<pre>
<code>
x = 5;
y = 6;
z = x + y;
</code>
</pre>
```

Result:

```
x = 5;
y = 6;
z = x + y;
```

Try it Yourself »

HTML <var> For Variables

The HTML `<var>` element is used to define a variable in programming or in a mathematical expression. The content inside is typically displayed in italic.

Example

Define some text as variables in a document:

```
<p>The area of a triangle is: 1/2 x <var>b</var> x <var>h</var>,
where <var>b</var> is the base, and <var>h</var> is the vertical
height.</p>
```

Result:

The area of a triangle is: 1/2 x *b* x *h*, where *b* is the base, and *h* is the vertical height.

Try it Yourself »

Chapter Summary

- The `<kbd>` element defines keyboard input
- The `<samp>` element defines sample output from a computer program
- The `<code>` element defines a piece of computer code
- The `<var>` element defines a variable in programming or in a mathematical expression
- The `<pre>` element defines preformatted text

HTML Exercises

Exercise:

Define the text "var person;" as programming code.

```
<p>Code example: □var person;□</p>
```

Start the Exercise

HTML Computer Code Elements

Tag	Description
<code>	Defines programming code
<kbd>	Defines keyboard input
<samp>	Defines computer output
<var>	Defines a variable
<pre>	Defines preformatted text

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Semantic Elements

Semantic elements = elements with a meaning.

What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of **non-semantic** elements: `<div>` and `<span>` - Tells nothing about its content.

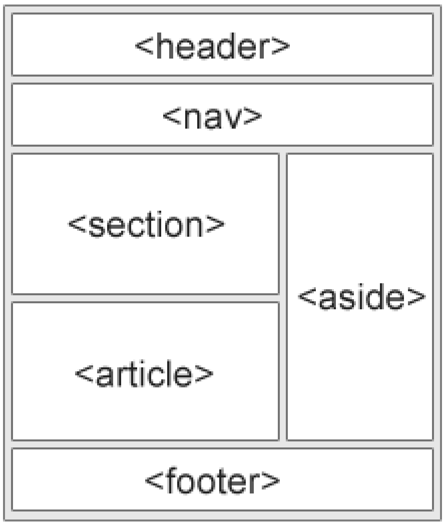
Examples of **semantic** elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.

Semantic Elements in HTML

Many web sites contain HTML code like: <div id="nav"> <div class="header"> <div id="footer"> to indicate navigation, header, and footer.

In HTML there are some semantic elements that can be used to define different parts of a web page:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



HTML <section> Element

The `<section>` element defines a section in a document.

According to W3C's HTML documentation: "A section is a thematic grouping of content, typically with a heading."

Examples of where a `<section>` element can be used:

- Chapters
- Introduction
- News items
- Contact information

A web page could normally be split into sections for introduction, content, and contact information.

Example

Two sections in a document:

```
<section>
<h1>WWF</h1>
<p>The World Wide Fund for Nature (WWF) is an international
organization working on issues regarding the conservation, research and
restoration of the environment, formerly named the World Wildlife Fund.
WWF was founded in 1961.</p>
</section>

<section>
<h1>WWF's Panda symbol</h1>
<p>The Panda has become the symbol of WWF. The well-known panda logo of
WWF originated from a panda named Chi Chi that was transferred from the
Beijing Zoo to the London Zoo in the same year of the establishment of
WWF.</p>
</section>
```

Try it Yourself »

HTML <article> Element

The `<article>` element specifies independent, self-contained content.

An article should make sense on its own, and it should be possible to distribute it independently from the rest of the web site.

Examples of where the `<article>` element can be used:

- Forum posts
- Blog posts
- User comments
- Product cards
- Newspaper articles

Example

Three articles with independent, self-contained content:

```
<article>
<h2>Google Chrome</h2>
<p>Google Chrome is a web browser developed by Google, released in
2008. Chrome is the world's most popular web browser today!</p>
```

```
</article>

<article>
<h2>Mozilla Firefox</h2>
<p>Mozilla Firefox is an open-source web browser developed by Mozilla.
Firefox has been the second most popular web browser since January,
2018.</p>
</article>

<article>
<h2>Microsoft Edge</h2>
<p>Microsoft Edge is a web browser developed by Microsoft, released in
2015. Microsoft Edge replaced Internet Explorer.</p>
</article>
```

Try it Yourself »

Example 2

Use CSS to style the <article> element:

```
<html>
<head>
<style>
.all-browsers {
  margin: 0;
  padding: 5px;
  background-color: lightgray;
}

.all-browsers > h1, .browser {
  margin: 10px;
  padding: 5px;
}

.browser {
  background: white;
}

.browser > h2, p {
  margin: 4px;
  font-size: 90%;
}
</style>
</head>
<body>

<article class="all-browsers">
```

```
  <h1>Most Popular Browsers</h1>
  <article class="browser">
    <h2>Google Chrome</h2>
    <p>Google Chrome is a web browser developed by Google, released in
2008. Chrome is the world's most popular web browser today!</p>
  </article>
  <article class="browser">
    <h2>Mozilla Firefox</h2>
    <p>Mozilla Firefox is an open-source web browser developed by
Mozilla. Firefox has been the second most popular web browser since
January, 2018.</p>
  </article>
  <article class="browser">
    <h2>Microsoft Edge</h2>
    <p>Microsoft Edge is a web browser developed by Microsoft, released
in 2015. Microsoft Edge replaced Internet Explorer.</p>
  </article>
</article>

</body>
</html>
```

Try it Yourself »

Nesting <article> in <section> or Vice Versa?

The `<article>` element specifies independent, self-contained content.

The `<section>` element defines section in a document.

Can we use the definitions to decide how to nest those elements? No, we cannot!

So, you will find HTML pages with `<section>` elements containing `<article>` elements, and `<article>` elements containing `<section>` elements.

HTML <header> Element

The `<header>` element represents a container for introductory content or a set of navigational links.

A `<header>` element typically contains:

- one or more heading elements (<h1> - <h6>)
- logo or icon
- authorship information

Note: You can have several <header> elements in one HTML document. However, <header> cannot be placed within a <footer>, <address> or another <header> element.

Example

A header for an <article>:

```
<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural
environment,
  and build a future in which humans live in harmony with nature.</p>
</article>
```

Try it Yourself »

HTML <footer> Element

The <footer> element defines a footer for a document or section.

A <footer> element typically contains:

- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents

You can have several <footer> elements in one document.

Example

A footer section in a document:

```
<footer>
  <p>Author: Hege Refsnes</p>
  <p><a href="mailto:hege@example.com">hege@example.com</a></p>
</footer>
```

Try it Yourself »

HTML <nav> Element

The `<nav>` element defines a set of navigation links.

Notice that NOT all links of a document should be inside a `<nav>` element. The `<nav>` element is intended only for major blocks of navigation links.

Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

Example

A set of navigation links:

```
<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/jquery/">jQuery</a>
</nav>
```

Try it Yourself »

HTML <aside> Element

The `<aside>` element defines some content aside from the content it is placed in (like a sidebar).

The `<aside>` content should be indirectly related to the surrounding content.

Example

Display some content aside from the content it is placed in:

```
<p>My family and I visited The Epcot center this summer. The weather
was nice, and Epcot was amazing! I had a great summer together with my
family!</p>

<aside>
<h4>Epcot Center</h4>
<p>Epcot is a theme park at Walt Disney World Resort featuring exciting
attractions, international pavilions, award-winning fireworks and
seasonal special events.</p>
</aside>
```

Try it Yourself »

Example 2

Use CSS to style the <aside> element:

```
<html>
<head>
<style>
aside {
  width: 30%;
  padding-left: 15px;
  margin-left: 15px;
  float: right;
  font-style: italic;
  background-color: lightgray;
}
</style>
</head>
<body>

<p>My family and I visited The Epcot center this summer. The weather
was nice, and Epcot was amazing! I had a great summer together with my
family!</p>

<aside>
<p>The Epcot center is a theme park at Walt Disney World Resort
featuring exciting attractions, international pavilions, award-winning
fireworks and seasonal special events.</p>
</aside>

<p>My family and I visited The Epcot center this summer. The weather
was nice, and Epcot was amazing! I had a great summer together with my
family!</p>
<p>My family and I visited The Epcot center this summer. The weather
was nice, and Epcot was amazing! I had a great summer together with my
family!</p>

</body>
</html>
```

Try it Yourself »

HTML <figure> and <figcaption> Elements

The `<figure>` tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

The `<figcaption>` tag defines a caption for a `<figure>` element.
The `<figcaption>` element can be placed as the first or as the last child of a `<figure>` element.

The `<img>` element defines the actual image/illustration.

Example

```
<figure>
  <img src="pic_trulli.jpg" alt="Trulli">
  <figcaption>Fig1. - Trulli, Puglia, Italy.</figcaption>
</figure>
```

Try it Yourself »

Why Semantic Elements?

According to the W3C: "A semantic Web allows data to be shared and reused across applications, enterprises, and communities."

Semantic Elements in HTML

Below is a list of some of the semantic elements in HTML.

Tag	Description
<article>	Defines independent, self-contained content
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<figcaption>	Defines a caption for a <figure> element
<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

<footer>	Defines a footer for a document or section
<header>	Specifies a header for a document or section
<main>	Specifies the main content of a document
<mark>	Defines marked/highlighted text
<nav>	Defines navigation links
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Style Guide

A consistent, clean, and tidy HTML code makes it easier for others to read and understand your code.

Here are some guidelines and tips for creating good HTML code.

Always Declare Document Type

Always declare the document type as the first line in your document.

The correct document type for HTML is:

```
<!DOCTYPE html>
```

Use Lowercase Element Names

HTML allows mixing uppercase and lowercase letters in element names.

However, we recommend using lowercase element names, because:

- Mixing uppercase and lowercase names looks bad
- Developers normally use lowercase names
- Lowercase looks cleaner
- Lowercase is easier to write

Good:

```
<body>
<p>This is a paragraph.</p>
</body>
```

Bad:

```
<BODY>
<P>This is a paragraph.</P>
</BODY>
```

Close All HTML Elements

In HTML, you do not have to close all elements (for example the `<p>` element).

However, we strongly recommend closing all HTML elements, like this:

Good:

```
<section>
  <p>This is a paragraph.</p>
  <p>This is a paragraph.</p>
</section>
```

Bad:

```
<section>
  <p>This is a paragraph.
  <p>This is a paragraph.
</section>
```

Use Lowercase Attribute Names

HTML allows mixing uppercase and lowercase letters in attribute names.

However, we recommend using lowercase attribute names, because:

- Mixing uppercase and lowercase names looks bad
- Developers normally use lowercase names
- Lowercase looks cleaner
- Lowercase is easier to write

Good:

```
<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

Bad:

```
<a HREF="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

Always Quote Attribute Values

HTML allows attribute values without quotes.

However, we recommend quoting attribute values, because:

- Developers normally quote attribute values
- Quoted values are easier to read
- You MUST use quotes if the value contains spaces

Good:

```
<table class="striped">
```

Bad:

```
<table class=striped>
```

Very bad:

This will not work, because the value contains spaces:

```
<table class=table striped>
```

Always Specify alt, width, and height for Images

Always specify the `alt` attribute for images. This attribute is important if the image for some reason cannot be displayed.

Also, always define the `width` and `height` of images. This reduces flickering, because the browser can reserve space for the image before loading.

Good:

```
<img src="html5.gif" alt="HTML5" style="width:128px;height:128px">
```

Bad:

```
<img src="html5.gif">
```

Spaces and Equal Signs

HTML allows spaces around equal signs. But space-less is easier to read and groups entities better together.

Good:

```
<link rel="stylesheet" href="styles.css">
```

Bad:

```
<link rel = "stylesheet" href = "styles.css">
```

Avoid Long Code Lines

When using an HTML editor, it is NOT convenient to scroll right and left to read the HTML code.

Try to avoid too long code lines.

Blank Lines and Indentation

Do not add blank lines, spaces, or indentations without a reason.

For readability, add blank lines to separate large or logical code blocks.

For readability, add two spaces of indentation. Do not use the tab key.

Good:

```
<body>

<h1>Famous Cities</h1>

<h2>Tokyo</h2>
```

```
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
and the most populous metropolitan area in the world.</p>

<h2>London</h2>
<p>London is the capital city of England. It is the most populous city
in the United Kingdom.</p>

<h2>Paris</h2>
<p>Paris is the capital of France. The Paris area is one of the largest
population centers in Europe.</p>

</body>
```

Bad:

```
<body>
<h1>Famous Cities</h1>
<h2>Tokyo</h2><p>Tokyo is the capital of Japan, the center of the
Greater Tokyo Area, and the most populous metropolitan area in the
world.</p>
<h2>London</h2><p>London is the capital city of England. It is the most
populous city in the United Kingdom.</p>
<h2>Paris</h2><p>Paris is the capital of France. The Paris area is one
of the largest population centers in Europe.</p>
</body>
```

Good Table Example:

```
<table>
  <tr>
    <th>Name</th>
    <th>Description</th>
  </tr>
  <tr>
    <td>A</td>
    <td>Description of A</td>
  </tr>
  <tr>
    <td>B</td>
    <td>Description of B</td>
  </tr>
</table>
```

Good List Example:

```
<ul>
  <li>London</li>
  <li>Paris</li>
  <li>Tokyo</li>
</ul>
```

Never Skip the <title> Element

The `<title>` element is required in HTML.

The contents of a page title is very important for search engine optimization (SEO)! The page title is used by search engine algorithms to decide the order when listing pages in search results.

The `<title>` element:

- defines a title in the browser toolbar
- provides a title for the page when it is added to favorites
- displays a title for the page in search-engine results

So, try to make the title as accurate and meaningful as possible:

```
<title>HTML Style Guide and Coding Conventions</title>
```

Omitting <html> and <body>?

An HTML page will validate without the `<html>` and `<body>` tags:

Example

```
<!DOCTYPE html>
<head>
  <title>Page Title</title>
</head>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

Try it Yourself »

However, we strongly recommend to always add the `<html>` and `<body>` tags!

Omitting `<body>` can produce errors in older browsers.

Omitting `<html>` and `<body>` can also crash DOM and XML software.

Omitting <head>?

The HTML <head> tag can also be omitted.

Browsers will add all elements before `<body>`, to a default `<head>` element.

Example

```
<!DOCTYPE html>
<html>
<title>Page Title</title>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Try it Yourself »

However, we recommend using the `<head>` tag.

Close Empty HTML Elements?

In HTML, it is optional to close empty elements.

Allowed:

```
<meta charset="utf-8">
```

Also Allowed:

```
<meta charset="utf-8" />
```

If you expect XML/XHTML software to access your page, keep the closing slash (/), because it is required in XML and XHTML.

Add the lang Attribute

You should always include the `lang` attribute inside the `<html>` tag, to declare the language of the Web page. This is meant to assist search engines and browsers.

Example

```
<!DOCTYPE html>
<html lang="en-us">
<head>
  <title>Page Title</title>
</head>
<body>
```

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Try it Yourself »

Meta Data

To ensure proper interpretation and correct search engine indexing, both the language and the character encoding `<meta charset="charset">` should be defined as early as possible in an HTML document:

```
<!DOCTYPE html>
<html lang="en-us">
<head>
  <meta charset="UTF-8">
  <title>Page Title</title>
</head>
```

Setting The Viewport

The viewport is the user's visible area of a web page. It varies with the device - it will be smaller on a mobile phone than on a computer screen.

You should include the following `<meta>` element in all your web pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

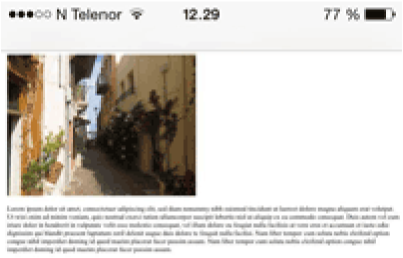
This gives the browser instructions on how to control the page's dimensions and scaling.

The `width=device-width` part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

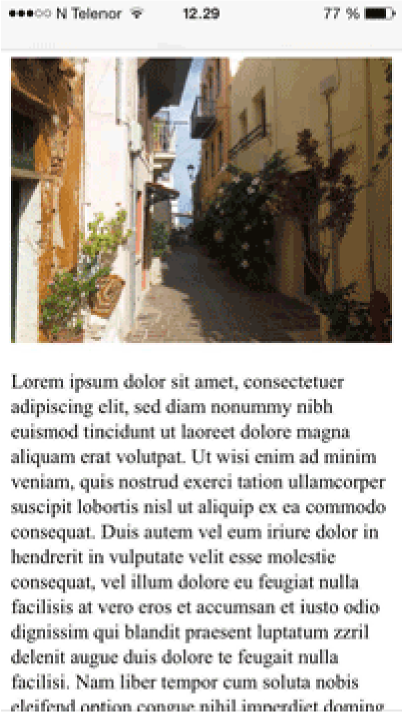
The `initial-scale=1.0` part sets the initial zoom level when the page is first loaded by the browser.

Here is an example of a web page *without* the viewport meta tag, and the same web page *with* the viewport meta tag:

Tip: If you are browsing this page with a phone or a tablet, you can click on the two links below to see the difference.



[Without the viewport meta tag]



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming

[With the viewport meta tag]

HTML Comments

Short comments should be written on one line, like this:

```
<!-- This is a comment -->
```

Comments that spans more than one line, should be written like this:

```
<!--
  This is a long comment example. This is a long comment example.
  This is a long comment example. This is a long comment example.
-->
```

Long comments are easier to observe if they are indented with two spaces.

Using Style Sheets

Use simple syntax for linking to style sheets (the `type` attribute is not necessary):

```
<link rel="stylesheet" href="styles.css">
```

Short CSS rules can be written compressed, like this:

```
p.intro {font-family:Verdana;font-size:16em;}
```

Long CSS rules should be written over multiple lines:

```
body {
  background-color: lightgrey;
  font-family: "Arial Black", Helvetica, sans-serif;
  font-size: 16em;
  color: black;
}
```

- Place the opening bracket on the same line as the selector
- Use one space before the opening bracket
- Use two spaces of indentation
- Use semicolon after each property-value pair, including the last
- Only use quotes around values if the value contains spaces
- Place the closing bracket on a new line, without leading spaces

Loading JavaScript in HTML

Use simple syntax for loading external scripts (the `type` attribute is not necessary):

```
<script src="myscript.js">
```

Accessing HTML Elements with JavaScript

Using "untidy" HTML code can result in JavaScript errors.

These two JavaScript statements will produce different results:

Example

```
getElementById("Demo").innerHTML = "Hello";

getElementById("demo").innerHTML = "Hello";
```

Try it Yourself »

Visit the JavaScript Style Guide.

Use Lower Case File Names

Some web servers (Apache, Unix) are case sensitive about file names: "london.jpg" cannot be accessed as "London.jpg".

Other web servers (Microsoft, IIS) are not case sensitive: "london.jpg" can be accessed as "London.jpg".

If you use a mix of uppercase and lowercase, you have to be aware of this.

If you move from a case-insensitive to a case-sensitive server, even small errors will break your web!

To avoid these problems, always use lowercase file names!

File Extensions

HTML files should have a **.html** extension (**.htm** is allowed).

CSS files should have a **.css** extension.

JavaScript files should have a **.js** extension.

Differences Between .htm and .html?

There is no difference between the .htm and .html file extensions!

Both will be treated as HTML by any web browser and web server.

Default Filenames

When a URL does not specify a filename at the end (like "https://www.w3schools.com/"), the server just adds a default filename, such as "index.html", "index.htm", "default.html", or "default.htm".

If your server is configured only with "index.html" as the default filename, your file must be named "index.html", and not "default.html".

However, servers can be configured with more than one default filename; usually you can set up as many default filenames as you want.

HTML Entities

Reserved characters in HTML must be replaced with entities:

- < (less than) = **&lt;**
- > (greather than) = **&gt;**

HTML Character Entities

Some characters are reserved in HTML.

If you use the less than (<) or greater than (>) signs in your HTML text, the browser might mix them with tags.

Entity names or entity numbers can be used to display reserved HTML characters.

Entity names look like this:

&*entity_name*;

Entity numbers look like this:

&#*entity_number*;

To display a less than sign (<) we must write: **&lt;** or **&#60;**

Entity names are easier to remember than entity numbers.

Non-breaking Space

A commonly used HTML entity is the non-breaking space: **&nbsp;**

A non-breaking space is a space that will not break into a new line.

Two words separated by a non-breaking space will stick together (not break into a new line). This is handy when breaking the words might be disruptive.

Examples:

- § 10
- 10 km/h
- 10 PM

Another common use of the non-breaking space is to prevent browsers from truncating spaces in HTML pages.

If you write 10 spaces in your text, the browser will remove 9 of them. To add real spaces to your text, you can use the **&nbsp;** character entity.

The non-breaking hyphen (&#8209;) is used to define a hyphen character (‑) that does not break into a new line.

Some Useful HTML Character Entities

Result	Description	Name	Number	
	non-breaking space	&nbsp;	&#160;	Try it »
<	less than	&lt;	&#60;	Try it »
>	greater than	&gt;	&#62;	Try it »
&	ampersand	&amp;	&#38;	Try it »

"	double quotation mark	&quot;	&#34;	Try it »
'	single quotation mark	&apos;	&#39;	Try it »
¢	cent	&cent;	&#162;	Try it »
£	pound	&pound;	&#163;	Try it »
¥	yen	&yen;	&#165;	Try it »
€	euro	&euro;	&#8364;	Try it »
©	copyright	&copy;	&#169;	Try it »
®	trademark	&reg;	&#174;	Try it »

Note

Entity names are case sensitive.

Combining Diacritical Marks

A diacritical mark is a "glyph" added to a letter.

Some diacritical marks, like grave (̀) and acute (́) are called accents.

Diacritical marks can be used in combination with alphanumeric characters to produce a character that is not present in the character set (encoding) used in the page.

Here are some examples:

Mark	Character	Construct	Result	
̀	a	a&#768;	à	Try it »
́	a	a&#769;	á	Try it »
̂	a	a&#770;	â	Try it »
̃	a	a&#771;	ã	Try it »
̀	O	O&#768;	Ò	Try it »
́	O	O&#769;	Ó	Try it »
̂	O	O&#770;	Ô	Try it »

~	O	O&#771;	Õ	Try it »

There are more examples in the next chapter.

HTML Symbols

Symbols or letters that are not present on your keyboard can be added to HTML using entities.

HTML Symbol Entities

HTML entities were described in the previous chapter.

Many mathematical, technical, and currency symbols, are not present on a normal keyboard.

To add such symbols to an HTML page, you can use the entity name or the entity number (a decimal or a hexadecimal reference) for the symbol:

Example

Display the euro sign:

```
<p>I will display &euro;</p>
<p>I will display &#8364;</p>
<p>I will display &#x20AC;</p>
```

Will display as:

I will display €
I will display €
I will display €

Try it Yourself »

Some Mathematical Symbols Supported by HTML

Char	Number	Entity	Description

∀	&#8704;	&forall;	For all	Try it »
∂	&#8706;	&part;	Partial differential	Try it »
∃	&#8707;	&exist;	There exists	Try it »
∅	&#8709;	&empty;	Empty sets	Try it »
∇	&#8711;	&nabla;	Nabla	Try it »
∈	&#8712;	&isin;	Element of	Try it »
∉	&#8713;	&notin;	Not an element of	Try it »
∋	&#8715;	&ni;	Contains as member	Try it »
∏	&#8719;	&prod;	N-ary product	Try it »
∑	&#8721;	&sum;	N-ary summation	Try it »

Full Math Symbols Reference

Some Greek Letters Supported by HTML

Char	Number	Entity	Description	
Α	&#913;	&Alpha;	GREEK ALPHA	Try it »
Β	&#914;	&Beta;	GREEK BETA	Try it »
Γ	&#915;	&Gamma;	GREEK GAMMA	Try it »
Δ	&#916;	&Delta;	GREEK DELTA	Try it »
Ε	&#917;	&Epsilon;	GREEK EPSILON	Try it »
Ζ	&#918;	&Zeta;	GREEK ZETA	Try it »

Full Greek Reference

Some Other Entities Supported by HTML

Char	Number	Entity	Description

©	&#169;	&copy;	COPYRIGHT	Try it »
®	&#174;	&reg;	REGISTERED	Try it »
€	&#8364;	&euro;	EURO SIGN	Try it »
™	&#8482;	&trade;	TRADEMARK	Try it »
←	&#8592;	&larr;	LEFT ARROW	Try it »
↑	&#8593;	&uarr;	UP ARROW	Try it »
→	&#8594;	&rarr;	RIGHT ARROW	Try it »
↓	&#8595;	&darr;	DOWN ARROW	Try it »
♠	&#9824;	&spades;	SPADE	Try it »
♣	&#9827;	&clubs;	CLUB	Try it »

♥	&#9829;	&hearts;	HEART	Try it »
♦	&#9830;	&diams;	DIAMOND	Try it »

Full Currency Reference

Full Arrows Reference

Full Symbols Reference

Using Emojis in HTML

Emojis are characters from the UTF-8 character set: 😄 😍 💗

Emoji	Value	
🗻 🗻	&#128507;	Try it »
🗼 🗼	&#128508;	Try it »
🗽 🗽	&#128509;	Try it »
🗾 🗾	&#128510;	Try it »

🗿 🗿	&#128511;	Try it »
😀 😀	&#128512;	Try it »
😁 😁	&#128513;	Try it »
😂 😂	&#128514;	Try it »
😃 😃	&#128515;	Try it »
😄 😄	&#128516;	Try it »
😅 😅	&#128517;	Try it »

Full HTML Emoji Reference

HTML Emojis Examples

🚀 🚁 🚂 🚃 🚄 🚀 🚁 🚂 🚃 🚄

HTML Emoji Transport Symbols

💺💻💼📀💾 💺💻💼📀💾

[HTML Emoji Office Symbols]

👮👯👰👱👲 👮👯👰👱👲

[HTML Emoji People Symbols]

🐂🐃🐄🐅🐆 🐂🐃🐄🐅🐆

[HTML Emoji Animals Symbols]

What are Emojis?

Emojis look like images, or icons, but they are not.

They are letters (characters) from the UTF-8 (Unicode) character set.

UTF-8 covers almost all of the characters and symbols in the world.

The HTML charset Attribute

To display an HTML page correctly, a web browser must know the character set used in the page.

This is specified in the `<meta>` tag:

```
<meta charset="UTF-8">
```

If not specified, UTF-8 is the default character set in HTML.

UTF-8 Characters

Many UTF-8 characters cannot be typed on a keyboard, but they can always be displayed using numbers (called entity numbers):

- A is 65
- B is 66
- C is 67

Example

```
<!DOCTYPE html>
<html>
<meta charset="UTF-8">
<body>

<p>I will display A B C</p>
<p>I will display &#65; &#66; &#67;</p>

</body>
</html>
```

Try it Yourself »

Example Explained

The `<meta charset="UTF-8">` element defines the character set.

The characters A, B, and C, are displayed by the numbers 65, 66, and 67.

To let the browser understand that you are displaying a character, you must start the entity number with &# and end it with ; (semicolon).

Emoji Characters

Emojis are also characters from the UTF-8 alphabet:

- 😄 😄 is 128516
- 😍 😍 is 128525
- 💗 💗 is 128151

Example

```
<!DOCTYPE html>
<html>
<meta charset="UTF-8">
<body>

<h1>My First Emoji</h1>

<p>&#128512;</p>

</body>
</html>
```

Try it Yourself »

Since Emojis are characters, they can be copied, displayed, and sized just like any other character in HTML.

Example

```
<!DOCTYPE html>
<html>
<meta charset="UTF-8">
<body>

<h1>Sized Emojis</h1>

<p style="font-size:48px">
&#128512; &#128516; &#128525; &#128151;
</p>

</body>
</html>
```

Try it Yourself »

HTML Encoding (Character Sets)

To display an HTML page correctly, a web browser must know which character set to use.

The HTML charset Attribute

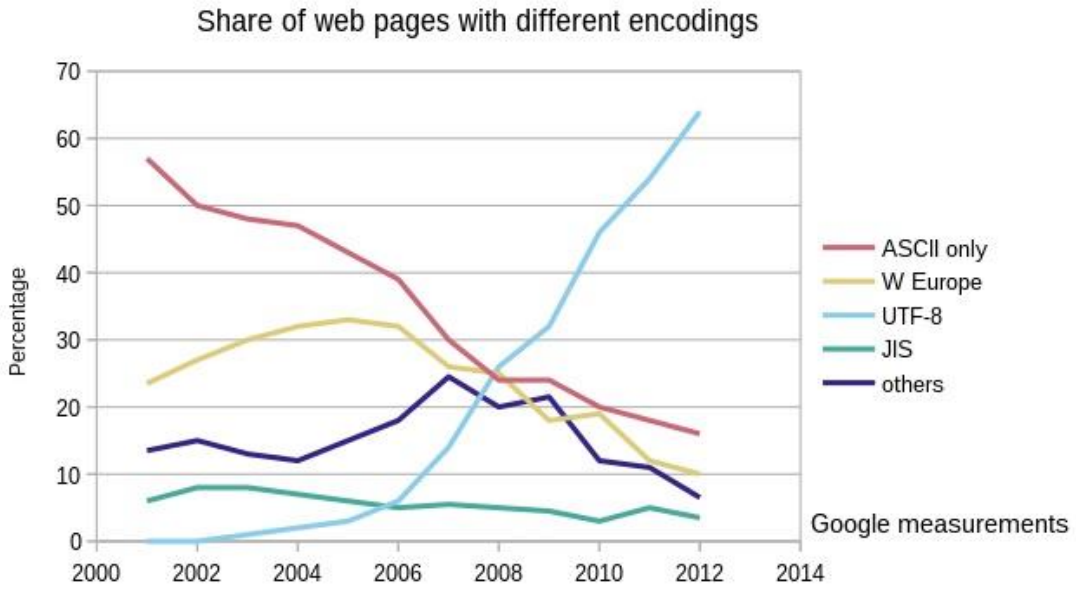
The character set is specified in the `<meta>` tag:

Example

```
<meta charset="UTF-8">
```

The HTML5 specification encourages web developers to use the UTF-8 character set.

UTF-8 covers almost all of the characters and symbols in the world!



[Full UTF-8 Reference](#)

The ASCII Character Set

ASCII was the first character encoding standard for the web. It defined 128 different characters that could be used on the internet:

- English letters (A-Z)
- Numbers (0-9)
- Special characters like ! $ + - ( ) @ < >.

The ANSI Character Set

ANSI (Windows-1252) was the original Windows character set:

- Identical to ASCII for the first 127 characters
- Special characters from 128 to 159
- Identical to UTF-8 from 160 to 255

```
<meta charset="Windows-1252">
```

The ISO-8859-1 Character Set

ISO-8859-1 was the default character set for HTML 4. This character set supported 256 different character codes. HTML 4 also supported UTF-8.

- Identical to ASCII for the first 127 characters
- Does not use the characters from 128 to 159
- Identical to ANSI and UTF-8 from 160 to 255

HTML 4 Example

```
<meta http-equiv="Content-Type" content="text/html;charset=ISO-8859-1">
```

HTML 5 Example

```
<meta charset="ISO-8859-1">
```

The UTF-8 Character Set

- is identical to ASCII for the values from 0 to 127
- Does not use the characters from 128 to 159
- Identical to ANSI and 8859-1 from 160 to 255
- Continues from the value 256 to 10 000 characters

```
<meta charset="UTF-8">
```

Full HTML Character Set Reference.

Differences Between Character Sets

The following table displays the differences between the character sets described above:

Numb	ASCII	ANSI	8859	UTF-8	Description
32					space
33	!	!	!	!	exclamation mark
34	"	"	"	"	quotation mark
35	#	#	#	#	number sign
36	$	$	$	$	dollar sign

37	%	%	%	%	percent sign
38	&	&	&	&	ampersand
39	'	'	'	'	apostrophe
40	(	(	(	(	left parenthesis
41	)	)	)	)	right parenthesis
42	*	*	*	*	asterisk
43	+	+	+	+	plus sign
44	,	,	,	,	comma
45	-	-	-	-	hyphen-minus
46	.	.	.	.	full stop
47	/	/	/	/	solidus
48	0	0	0	0	digit zero

49	1	1	1	1	digit one
50	2	2	2	2	digit two
51	3	3	3	3	digit three
52	4	4	4	4	digit four
53	5	5	5	5	digit five
54	6	6	6	6	digit six
55	7	7	7	7	digit seven
56	8	8	8	8	digit eight
57	9	9	9	9	digit nine
58	:	:	:	:	colon
59	;	;	;	;	semicolon
60	<	<	<	<	less than

61	=	=	=	=	equals sign
62	>	>	>	>	greater than
63	?	?	?	?	question mark
64	@	@	@	@	commercial at
65	A	A	A	A	Latin A
66	B	B	B	B	Latin B
67	C	C	C	C	Latin C
68	D	D	D	D	Latin D
69	E	E	E	E	Latin E
70	F	F	F	F	Latin F
71	G	G	G	G	Latin G
72	H	H	H	H	Latin H

73	I	I	I	I	Latin I
74	J	J	J	J	Latin J
75	K	K	K	K	Latin K
76	L	L	L	L	Latin L
77	M	M	M	M	Latin M
78	N	N	N	N	Latin N
79	O	O	O	O	Latin O
80	P	P	P	P	Latin P
81	Q	Q	Q	Q	Latin Q
82	R	R	R	R	Latin R
83	S	S	S	S	Latin S
84	T	T	T	T	Latin T

85	U	U	U	U	Latin U
86	V	V	V	V	Latin V
87	W	W	W	W	Latin W
88	X	X	X	X	Latin X
89	Y	Y	Y	Y	Latin Y
90	Z	Z	Z	Z	Latin Z
91	[	[	[	[	left square bracket
92	\	\	\	\	reverse solidus
93	]	]	]	]	right square bracket
94	^	^	^	^	circumflex accent
95	_	_	_	_	low line
96	`	`	`	`	grave accent

97	a	a	a	a	Latin small a
98	b	b	b	b	Latin small b
99	c	c	c	c	Latin small c
100	d	d	d	d	Latin small d
101	e	e	e	e	Latin small e
102	f	f	f	f	Latin small f
103	g	g	g	g	Latin small g
104	h	h	h	h	Latin small h
105	i	i	i	i	Latin small i
106	j	j	j	j	Latin small j
107	k	k	k	k	Latin small k
108	l	l	l	l	Latin small l

109	m	m	m	m	Latin small m
110	n	n	n	n	Latin small n
111	o	o	o	o	Latin small o
112	p	p	p	p	Latin small p
113	q	q	q	q	Latin small q
114	r	r	r	r	Latin small r
115	s	s	s	s	Latin small s
116	t	t	t	t	Latin small t
117	u	u	u	u	Latin small u
118	v	v	v	v	Latin small v
119	w	w	w	w	Latin small w
120	x	x	x	x	Latin small x

121	y	y	y	y	Latin small y
122	z	z	z	z	Latin small z
123	{	{	{	{	left curly bracket
124	|	|	|	|	vertical line
125	}	}	}	}	right curly bracket
126	~	~	~	~	tilde
127	DEL				
128		€			euro sign
129		•	•	•	NOT USED
130		‚			single low-9 quotation mark
131		ƒ			Latin small f with hook
132		„			double low-9 quotation mark

133	…			horizontal ellipsis
134	†			dagger
135	‡			double dagger
136	ˆ			modifier letter circumflex accent
137	‰			per mille sign
138	Š			Latin S with caron
139	‹			single left-pointing angle quotation mark
140	Œ			Latin capital ligature OE
141	•	•	•	NOT USED
142	Ž			Latin Z with caron
143	•	•	•	NOT USED
144	•	•	•	NOT USED

145	‘	left single quotation mark
146	’	right single quotation mark
147	“	left double quotation mark
148	”	right double quotation mark
149	•	bullet
150	–	en dash
151	—	em dash
152	˜	small tilde
153	™	trade mark sign
154	š	Latin small s with caron
155	›	single right-pointing angle quotation mark
156	œ	Latin small ligature oe

157	•	•	•	NOT USED
158	ž			Latin small z with caron
159	Ÿ			Latin Y with diaeresis
160				no-break space
161	¡	¡	¡	inverted exclamation mark
162	¢	¢	¢	cent sign
163	£	£	£	pound sign
164	¤	¤	¤	currency sign
165	¥	¥	¥	yen sign
166	¦	¦	¦	broken bar
167	§	§	§	section sign
168	¨	¨	¨	diaeresis

169		©	©	©	copyright sign
170		ª	ª	ª	feminine ordinal indicator
171		«	«	«	left-pointing double angle quotation mark
172		¬	¬	¬	not sign
173					soft hyphen
174		®	®	®	registered sign
175		¯	¯	¯	macron
176		°	°	°	degree sign
177		±	±	±	plus-minus sign
178		²	²	²	superscript two
179		³	³	³	superscript three
180		´	´	´	acute accent

181	µ	µ	µ	micro sign
182	¶	¶	¶	pilcrow sign
183	·	·	·	middle dot
184	¸	¸	¸	cedilla
185	¹	¹	¹	superscript one
186	º	º	º	masculine ordinal indicator
187	»	»	»	right-pointing double angle quotation mark
188	¼	¼	¼	vulgar fraction one quarter
189	½	½	½	vulgar fraction one half
190	¾	¾	¾	vulgar fraction three quarters
191	¿	¿	¿	inverted question mark
192	À	À	À	Latin A with grave

193	Á	Á	Á	Latin A with acute
194	Â	Â	Â	Latin A with circumflex
195	Ã	Ã	Ã	Latin A with tilde
196	Ä	Ä	Ä	Latin A with diaeresis
197	Å	Å	Å	Latin A with ring above
198	Æ	Æ	Æ	Latin AE
199	Ç	Ç	Ç	Latin C with cedilla
200	È	È	È	Latin E with grave
201	É	É	É	Latin E with acute
202	Ê	Ê	Ê	Latin E with circumflex
203	Ë	Ë	Ë	Latin E with diaeresis
204	Ì	Ì	Ì	Latin I with grave

205	Í	Í	Í	Latin I with acute
206	Î	Î	Î	Latin I with circumflex
207	Ï	Ï	Ï	Latin I with diaeresis
208	Ð	Ð	Ð	Latin Eth
209	Ñ	Ñ	Ñ	Latin N with tilde
210	Ò	Ò	Ò	Latin O with grave
211	Ó	Ó	Ó	Latin O with acute
212	Ô	Ô	Ô	Latin O with circumflex
213	Õ	Õ	Õ	Latin O with tilde
214	Ö	Ö	Ö	Latin O with diaeresis
215	×	×	×	multiplication sign
216	Ø	Ø	Ø	Latin O with stroke

217		Ù	Ù	Ù	Latin U with grave
218		Ú	Ú	Ú	Latin U with acute
219		Û	Û	Û	Latin U with circumflex
220		Ü	Ü	Ü	Latin U with diaeresis
221		Ý	Ý	Ý	Latin Y with acute
222		Þ	Þ	Þ	Latin Thorn
223		ß	ß	ß	Latin small sharp s
224		à	à	à	Latin small a with grave
225		á	á	á	Latin small a with acute
226		â	â	â	Latin small a with circumflex
227		ã	ã	ã	Latin small a with tilde
228		ä	ä	ä	Latin small a with diaeresis

229		å	å	å	Latin small a with ring above
230		æ	æ	æ	Latin small ae
231		ç	ç	ç	Latin small c with cedilla
232		è	è	è	Latin small e with grave
233		é	é	é	Latin small e with acute
234		ê	ê	ê	Latin small e with circumflex
235		ë	ë	ë	Latin small e with diaeresis
236		ì	ì	ì	Latin small i with grave
237		í	í	í	Latin small i with acute
238		î	î	î	Latin small i with circumflex
239		ï	ï	ï	Latin small i with diaeresis
240		ð	ð	ð	Latin small eth

241		ñ	ñ	ñ	Latin small n with tilde
242		ò	ò	ò	Latin small o with grave
243		ó	ó	ó	Latin small o with acute
244		ô	ô	ô	Latin small o with circumflex
245		õ	õ	õ	Latin small o with tilde
246		ö	ö	ö	Latin small o with diaeresis
247		÷	÷	÷	division sign
248		ø	ø	ø	Latin small o with stroke
249		ù	ù	ù	Latin small u with grave
250		ú	ú	ú	Latin small u with acute
251		û	û	û	Latin small with circumflex
252		ü	ü	ü	Latin small u with diaeresis

253	ý	ý	ý	Latin small y with acute
254	þ	þ	þ	Latin small thorn
255	ÿ	ÿ	ÿ	Latin small y with diaeresis

HTML Uniform Resource Locators

A URL is another word for a web address.

A URL can be composed of words (e.g. w3schools.com), or an Internet Protocol (IP) address (e.g. 192.68.20.50).

Most people enter the name when surfing, because names are easier to remember than numbers.

URL - Uniform Resource Locator

Web browsers request pages from web servers by using a URL.

A Uniform Resource Locator (URL) is used to address a document (or other data) on the web.

A web address like https://www.w3schools.com/html/default.asp follows these syntax rules:

```
scheme://prefix.domain:port/path/filename
```

Explanation:

- **scheme** - defines the **type** of Internet service (most common is **http or https**)
- **prefix** - defines a domain **prefix** (default for http is **www**)
- **domain** - defines the Internet **domain name** (like w3schools.com)
- **port** - defines the **port number** at the host (default for http is **80**)
- **path** - defines a **path** at the server (If omitted: the root directory of the site)
- **filename** - defines the name of a document or resource

Common URL Schemes

The table below lists some common schemes:

Scheme	Short for	Used for
http	HyperText Transfer Protocol	Common web pages. Not encrypted
https	Secure HyperText Transfer Protocol	Secure web pages. Encrypted
ftp	File Transfer Protocol	Downloading or uploading files
file		A file on your computer

URL Encoding

URLs can only be sent over the Internet using the ASCII character-set. If a URL contains characters outside the ASCII set, the URL has to be converted.

URL encoding converts non-ASCII characters into a format that can be transmitted over the Internet.

URL encoding replaces non-ASCII characters with a "%" followed by hexadecimal digits.

URLs cannot contain spaces. URL encoding normally replaces a space with a plus (+) sign, or %20.

Try It Yourself

Hello Günter

Submit

If you click "Submit", the browser will URL encode the input before it is sent to the server.

A page at the server will display the received input.

Try some other input and click Submit again.

ASCII Encoding Examples

Your browser will encode input, according to the character-set used in your page.

The default character-set in HTML5 is UTF-8.

Character	From Windows-1252	From UTF-8
€	%80	%E2%82%AC
£	%A3	%C2%A3
©	%A9	%C2%A9
®	%AE	%C2%AE
À	%C0	%C3%80
Á	%C1	%C3%81
Â	%C2	%C3%82

Ã	%C3	%C3%83
Ä	%C4	%C3%84
Å	%C5	%C3%85

For a complete reference of all URL encodings, visit our URL Encoding Reference.

HTML Versus XHTML

XHTML is a stricter, more XML-based version of HTML.

What is XHTML?

- XHTML stands for E**X**tensible **H**yper**T**ext **M**arkup **L**anguage
- XHTML is a stricter, more XML-based version of HTML
- XHTML is HTML defined as an XML application
- XHTML is supported by all major browsers

Why XHTML?

XML is a markup language where all documents must be marked up correctly (be "well-formed").

XHTML was developed to make HTML more extensible and flexible to work with other data formats (such as XML). In addition, browsers ignore errors in HTML pages, and try to display the website even if it has some errors in the markup. So XHTML comes with a much stricter error handling.

If you want to study XML, please read our XML Tutorial.

The Most Important Differences from HTML

- <!DOCTYPE> is **mandatory**
- The xmlns attribute in <html> is **mandatory**
- <html>, <head>, <title>, and <body> are **mandatory**
- Elements must always be **properly nested**
- Elements must always be **closed**

- Elements must always be in **lowercase**
- Attribute names must always be in **lowercase**
- Attribute values must always be **quoted**
- Attribute minimization is **forbidden**

XHTML - <!DOCTYPE ....> Is Mandatory

An XHTML document must have an XHTML <!DOCTYPE> declaration.

The <html>, <head>, <title>, and <body> elements must also be present, and the xmlns attribute in <html> must specify the xml namespace for the document.

Example

Here is an XHTML document with a minimum of required tags:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Title of document</title>
</head>
<body>

  some content here...

</body>
</html>
```

XHTML Elements Must be Properly Nested

In XHTML, elements must always be properly nested within each other, like this:

Correct:

```
<b><i>Some text</i></b>
```

Wrong:

```
<b><i>Some text</b></i>
```

XHTML Elements Must Always be Closed

In XHTML, elements must always be closed, like this:

Correct:

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

Wrong:

```
<p>This is a paragraph
<p>This is another paragraph
```

XHTML Empty Elements Must Always be Closed

In XHTML, empty elements must always be closed, like this:

Correct:

```
A break: <br />
A horizontal rule: <hr />
An image: <img src="happy.gif" alt="Happy face" />
```

Wrong:

```
A break: <br>
A horizontal rule: <hr>
An image: <img src="happy.gif" alt="Happy face">
```

XHTML Elements Must be in Lowercase

In XHTML, element names must always be in lowercase, like this:

Correct:

```
<body>
<p>This is a paragraph</p>
</body>
```

Wrong:

```
<BODY>
<P>This is a paragraph</P>
</BODY>
```

XHTML Attribute Names Must be in Lowercase

In XHTML, attribute names must always be in lowercase, like this:

Correct:

```
<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

Wrong:

```
<a HREF="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

XHTML Attribute Values Must be Quoted

In XHTML, attribute values must always be quoted, like this:

Correct:

```
<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

Wrong:

```
<a href=https://www.w3schools.com/html/>Visit our HTML tutorial</a>
```

XHTML Attribute Minimization is Forbidden

In XHTML, attribute minimization is forbidden:

Correct:

```
<input type="checkbox" name="vehicle" value="car" checked="checked" />
<input type="text" name="lastname" disabled="disabled" />
```

Wrong:

```
<input type="checkbox" name="vehicle" value="car" checked />
<input type="text" name="lastname" disabled />
```

Validate HTML With The W3C Validator

Put your web address in the box below:

https://www.w3schools.com/html/html_validate.html

Validate the page

HTML Forms

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

Example

First name:
John
Last name:
Doe

Submit

Try it Yourself »

The <form> Element

The HTML `<form>` element is used to create an HTML form for user input:

```
<form>
.
form elements
.
</form>
```

The `<form>` element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

All the different form elements are covered in this chapter: HTML Form Elements.

The <input> Element

The HTML `<input>` element is the most used form element.

An `<input>` element can be displayed in many ways, depending on the `type` attribute.

Here are some examples:

Type	Description

<input type="text">	Displays a single-line text input field
<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="submit">	Displays a submit button (for submitting the form)
<input type="button">	Displays a clickable button

All the different input types are covered in this chapter: HTML Input Types.

Text Fields

The `<input type="text">` defines a single-line input field for text input.

Example

A form with input fields for text:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

First name:

□

Last name:

Note: The form itself is not visible. Also note that the default width of an input field is 20 characters.

The <label> Element

Notice the use of the `<label>` element in the example above.

The `<label>` tag defines a label for many form elements.

The `<label>` element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focuses on the input element.

The `<label>` element also helps users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the `<label>` element, it toggles the radio button/checkbox.

The `for` attribute of the `<label>` tag should be equal to the `id` attribute of the `<input>` element to bind them together.

Radio Buttons

The `<input type="radio">` defines a radio button.

Radio buttons let a user select ONE of a limited number of choices.

Example

A form with radio buttons:

```
<p>Choose your favorite Web language:</p>

<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaSc
ript">
  <label for="javascript">JavaScript</label>
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

Choose your favorite Web language:

○ HTML

○ CSS

○ JavaScript

Checkboxes

The `<input type="checkbox">` defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

Example

A form with checkboxes:

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

□ I have a bike

□ I have a car

□ I have a boat

The Submit Button

The `<input type="submit">` defines a button for submitting the form data to a form-handler.

The form-handler is typically a file on the server with a script for processing input data.

The form-handler is specified in the form's `action` attribute.

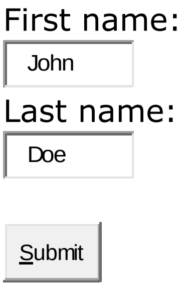
Example

A form with a submit button:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:



First name:

John

Last name:

Doe

Submit

The Name Attribute for <input>

Notice that each input field must have a `name` attribute to be submitted.

If the `name` attribute is omitted, the value of the input field will not be sent at all.

Example

This example will not submit the value of the "First name" input field:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" value="John"><br><br>
  <input type="submit" value="Submit">
</form>
```

Try it Yourself »

HTML Exercises

Exercise:

In the form below, add an input field with the type "button" and the value "OK".

```
<form>
<     >
</form>
```

Start the Exercise

HTML Form Attributes

This chapter describes the different attributes for the HTML `<form>` element.

The Action Attribute

The `action` attribute defines the action to be performed when the form is submitted.

Usually, the form data is sent to a file on the server when the user clicks on the submit button.

In the example below, the form data is sent to a file called "action_page.php". This file contains a server-side script that handles the form data:

Example

On submit, send form data to "action_page.php":

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

Try it Yourself »

Tip: If the `action` attribute is omitted, the action is set to the current page.

The Target Attribute

The `target` attribute specifies where to display the response that is received after submitting the form.

The `target` attribute can have one of the following values:

Value	Description
_blank	The response is displayed in a new window or tab
_self	The response is displayed in the current window
_parent	The response is displayed in the parent frame
_top	The response is displayed in the full body of the window
framename	The response is displayed in a named iframe

The default value is `_self` which means that the response will open in the current window.

Example

Here, the submitted result will open in a new browser tab:

```
<form action="/action_page.php" target="_blank">
```

Try it Yourself »

The Method Attribute

The `method` attribute specifies the HTTP method to be used when submitting the form data.

The form-data can be sent as URL variables (with `method="get"`) or as HTTP post transaction (with `method="post"`).

The default HTTP method when submitting form data is GET.

Example

This example uses the GET method when submitting the form data:

```
<form action="/action_page.php" method="get">
```

Try it Yourself »

Example

This example uses the POST method when submitting the form data:

```
<form action="/action_page.php" method="post">
```

Try it Yourself »

Notes on GET:

- Appends the form data to the URL, in name/value pairs
- NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)
- The length of a URL is limited (2048 characters)
- Useful for form submissions where a user wants to bookmark the result
- GET is good for non-secure data, like query strings in Google

Notes on POST:

- Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

Tip: Always use POST if the form data contains sensitive or personal information!

The Autocomplete Attribute

The `autocomplete` attribute specifies whether a form should have autocomplete on or off.

When autocomplete is on, the browser automatically complete values based on values that the user has entered before.

Example

A form with autocomplete on:

```
<form action="/action_page.php" autocomplete="on">
```

Try it Yourself »

The Novalidate Attribute

The `novalidate` attribute is a boolean attribute.

When present, it specifies that the form-data (input) should not be validated when submitted.

Example

A form with a novalidate attribute:

```
<form action="/action_page.php" novalidate>
```

Try it Yourself »

HTML Exercises

Exercise:

Add a submit button, and specify that the form should go to "/action_page.php".

```
<form □="/action_page.php">
Name: <input type="text" name="name">
<□>
</form>
```

Start the Exercise

List of All <form> Attributes

Attribute	Description
accept-charset	Specifies the character encodings used for form submission
action	Specifies where to send the form-data when a form is submitted
autocomplete	Specifies whether a form should have autocomplete on or off

enctype	Specifies how the form-data should be encoded when submitting it to the server (only for method="post")
method	Specifies the HTTP method to use when sending form-data
name	Specifies the name of the form
novalidate	Specifies that the form should not be validated when submitted
rel	Specifies the relationship between a linked resource and the current document
target	Specifies where to display the response that is received after submitting the form

HTML Form Elements

This chapter describes all the different HTML form elements.

The HTML <form> Elements

The HTML `<form>` element can contain one or more of the following form elements:

- `<input>`
- `<label>`
- `<select>`
- `<textarea>`
- `<button>`
- `<fieldset>`
- `<legend>`
- `<datalist>`
- `<output>`

- `<option>`
- `<optgroup>`

The <input> Element

One of the most used form elements is the `<input>` element.

The `<input>` element can be displayed in several ways, depending on the `type` attribute.

Example

```
<label for="fname">First name:</label>
<input type="text" id="fname" name="fname">
```

Try it Yourself »

All the different values of the `type` attribute are covered in the next chapter: HTML Input Types.

The <label> Element

The `<label>` element defines a label for several form elements.

The `<label>` element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.

The `<label>` element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the `<label>` element, it toggles the radio button/checkbox.

The `for` attribute of the `<label>` tag should be equal to the `id` attribute of the `<input>` element to bind them together.

The <select> Element

The `<select>` element defines a drop-down list:

Example

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Try it Yourself »

The <option> element defines an option that can be selected.

By default, the first item in the drop-down list is selected.

To define a pre-selected option, add the selected attribute to the option:

Example

```
<option value="fiat" selected>Fiat</option>
```

Try it Yourself »

Visible Values:

Use the size attribute to specify the number of visible values:

Example

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="3">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Try it Yourself »

Allow Multiple Selections:

Use the multiple attribute to allow the user to select more than one value:

Example

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="4" multiple>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Try it Yourself »

The <textarea> Element

The <textarea> element defines a multi-line input field (a text area):

Example

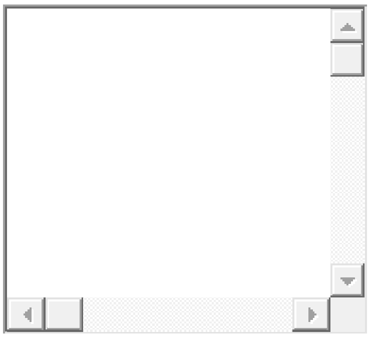
```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

Try it Yourself »

The rows attribute specifies the visible number of lines in a text area.

The cols attribute specifies the visible width of a text area.

This is how the HTML code above will be displayed in a browser:



You can also define the size of the text area by using CSS:

Example

```
<textarea name="message" style="width:200px; height:600px;">
The cat was playing in the garden.
</textarea>
```

Try it Yourself »

The <button> Element

The <button> element defines a clickable button:

Example

```
<button type="button" onclick="alert('Hello World!')">Click
Me!</button>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

Click Me!

Note: Always specify the `type` attribute for the button element. Different browsers may use different default types for the button element.

The <fieldset> and <legend> Elements

The `<fieldset>` element is used to group related data in a form.

The `<legend>` element defines a caption for the `<fieldset>` element.

Example

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

Personalia:First name:

John

Last name:

Doe

Submit

The <datalist> Element

The `<datalist>` element specifies a list of pre-defined options for an `<input>` element.

Users will see a drop-down list of the pre-defined options as they input data.

The `list` attribute of the `<input>` element, must refer to the `id` attribute of the `<datalist>` element.

Example

```
<form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Edge">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

Try it Yourself »

The <output> Element

The `<output>` element represents the result of a calculation (like one performed by a script).

Example

Perform a calculation and show the result in an `<output>` element:

```
<form action="/action_page.php"
  oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  0
  <input type="range"  id="a" name="a" value="50">
  100 +
  <input type="number" id="b" name="b" value="50">
  =
  <output name="x" for="a b"></output>
  <br><br>
  <input type="submit">
</form>
```

Try it Yourself »

HTML Exercises

Exercise:

In the form below, add an empty drop down list with the name "cars".

```
<form action="/action_page.php">
<□>
</□>
</form>
```

Start the Exercise

HTML Form Elements

Tag	Description
<form>	Defines an HTML form for user input
<input>	Defines an input control
<textarea>	Defines a multiline input control (text area)
<label>	Defines a label for an <input> element
<fieldset>	Groups related elements in a form
<legend>	Defines a caption for a <fieldset> element
<select>	Defines a drop-down list
<optgroup>	Defines a group of related options in a drop-down list
<option>	Defines an option in a drop-down list
<button>	Defines a clickable button

<datalist>	Specifies a list of pre-defined options for input controls
<output>	Defines the result of a calculation

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Input Types

This chapter describes the different types for the HTML `<input>` element.

HTML Input Types

Here are the different input types you can use in HTML:

- `<input type="button">`
- `<input type="checkbox">`
- `<input type="color">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="email">`
- `<input type="file">`
- `<input type="hidden">`
- `<input type="image">`
- `<input type="month">`
- `<input type="number">`
- `<input type="password">`
- `<input type="radio">`
- `<input type="range">`
- `<input type="reset">`
- `<input type="search">`
- `<input type="submit">`
- `<input type="tel">`
- `<input type="text">`
- `<input type="time">`
- `<input type="url">`
- `<input type="week">`

Tip: The default value of the `type` attribute is "text".

Input Type Text

`<input type="text">` defines a **single-line text input field**:

Example

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

First name:

Last name:

Input Type Password

`<input type="password">` defines a **password field**:

Example

```
<form>
  <label for="username">Username:</label><br>
  <input type="text" id="username" name="username"><br>
  <label for="pwd">Password:</label><br>
  <input type="password" id="pwd" name="pwd">
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

Username:

Password:

The characters in a password field are masked (shown as asterisks or circles).

Input Type Submit

`<input type="submit">` defines a button for **submitting** form data to a **form-handler**.

The form-handler is typically a server page with a script for processing input data.

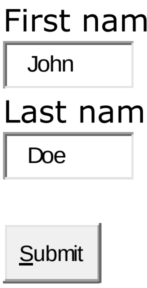
The form-handler is specified in the form's `action` attribute:

Example

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:



First name:

John

Last name:

Doe

Submit

If you omit the submit button's value attribute, the button will get a default text:

Example

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit">
</form>
```

Try it Yourself »

Input Type Reset

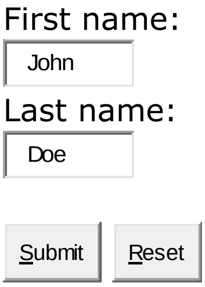
`<input type="reset">` defines a **reset button** that will reset all form values to their default values:

Example

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
  <input type="reset" value="Reset">
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:



First name:

John

Last name:

Doe

Submit Reset

If you change the input values and then click the "Reset" button, the form-data will be reset to the default values.

Input Type Radio

`<input type="radio">` defines a **radio button**.

Radio buttons let a user select ONLY ONE of a limited number of choices:

Example

```
<p>Choose your favorite Web language:</p>

<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaSc
ript">
```

```
  <label for="javascript">JavaScript</label>
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

○ HTML

○ CSS

○ JavaScript

Input Type Checkbox

`<input type="checkbox">` defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

Example

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

□ I have a bike

□ I have a car

□ I have a boat

Input Type Button

`<input type="button">` defines a **button**:

Example

```
<input type="button" onclick="alert('Hello World!')" value="Click Me!">
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

Input Type Color

The `<input type="color">` is used for input fields that should contain a color.

Depending on browser support, a color picker can show up in the input field.

Example

```
<form>
  <label for="favcolor">Select your favorite color:</label>
  <input type="color" id="favcolor" name="favcolor">
</form>
```

Try it Yourself »

Input Type Date

The `<input type="date">` is used for input fields that should contain a date.

Depending on browser support, a date picker can show up in the input field.

Example

```
<form>
  <label for="birthday">Birthday:</label>
  <input type="date" id="birthday" name="birthday">
</form>
```

Try it Yourself »

You can also use the `min` and `max` attributes to add restrictions to dates:

Example

```
<form>
  <label for="datemax">Enter a date before 1980-01-01:</label>
  <input type="date" id="datemax" name="datemax" max="1979-12-
31"><br><br>
  <label for="datemin">Enter a date after 2000-01-01:</label>
  <input type="date" id="datemin" name="datemin" min="2000-01-02">
</form>
```

Try it Yourself »

Input Type Datetime-local

The `<input type="datetime-local">` specifies a date and time input field, with no time zone.

Depending on browser support, a date picker can show up in the input field.

Example

```
<form>
  <label for="birthdaytime">Birthday (date and time):</label>
  <input type="datetime-local" id="birthdaytime" name="birthdaytime">
</form>
```

Try it Yourself »

Input Type Email

The `<input type="email">` is used for input fields that should contain an e-mail address.

Depending on browser support, the e-mail address can be automatically validated when submitted.

Some smartphones recognize the email type, and add ".com" to the keyboard to match email input.

Example

```
<form>
  <label for="email">Enter your email:</label>
  <input type="email" id="email" name="email">
</form>
```

Try it Yourself »

Input Type Image

The `<input type="image">` defines an image as a submit button.

The path to the image is specified in the `src` attribute.

Example

```
<form>
<input type="image" src="img_submit.gif" alt="Submit" width="48" height
```

```
="48">
</form>
```

Try it Yourself »

Input Type File

The `<input type="file">` defines a file-select field and a "Browse" button for file uploads.

Example

```
<form>
  <label for="myfile">Select a file:</label>
  <input type="file" id="myfile" name="myfile">
</form>
```

Try it Yourself »

Input Type Hidden

The `<input type="hidden">` defines a hidden input field (not visible to a user).

A hidden field lets web developers include data that cannot be seen or modified by users when a form is submitted.

A hidden field often stores what database record that needs to be updated when the form is submitted.

Note: While the value is not displayed to the user in the page's content, it is visible (and can be edited) using any browser's developer tools or "View Source" functionality. Do not use hidden inputs as a form of security!

Example

```
<form>
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <input type="hidden" id="custId" name="custId" value="3487">
  <input type="submit" value="Submit">
</form>
```

Try it Yourself »

Input Type Month

The `<input type="month">` allows the user to select a month and year.

Depending on browser support, a date picker can show up in the input field.

Example

```
<form>
  <label for="bdaymonth">Birthday (month and year):</label>
  <input type="month" id="bdaymonth" name="bdaymonth">
</form>
```

Try it Yourself »

Input Type Number

The `<input type="number">` defines a **numeric** input field.

You can also set restrictions on what numbers are accepted.

The following example displays a numeric input field, where you can enter a value from 1 to 5:

Example

```
<form>
  <label for="quantity">Quantity (between 1 and 5):</label>
  <input type="number" id="quantity" name="quantity" min="1" max="5">
</form>
```

Try it Yourself »

Input Restrictions

Here is a list of some common input restrictions:

Attribute	**Description**
checked	Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio")
disabled	Specifies that an input field should be disabled

max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)
size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

You will learn more about input restrictions in the next chapter.

The following example displays a numeric input field, where you can enter a value from 0 to 100, in steps of 10. The default value is 30:

Example

```
<form>
  <label for="quantity">Quantity:</label>
  <input type="number" id="quantity" name="quantity" min="0" max="100"
step="10" value="30">
</form>
```

Try it Yourself »

Input Type Range

The `<input type="range">` defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the `min`, `max`, and `step` attributes:

Example

```
<form>
  <label for="vol">Volume (between 0 and 50):</label>
  <input type="range" id="vol" name="vol" min="0" max="50">
</form>
```

Try it Yourself »

Input Type Search

The `<input type="search">` is used for search fields (a search field behaves like a regular text field).

Example

```
<form>
  <label for="gsearch">Search Google:</label>
  <input type="search" id="gsearch" name="gsearch">
</form>
```

Try it Yourself »

Input Type Tel

The `<input type="tel">` is used for input fields that should contain a telephone number.

Example

```
<form>
  <label for="phone">Enter your phone number:</label>
  <input type="tel" id="phone" name="phone" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">
</form>
```

Try it Yourself »

Input Type Time

The `<input type="time">` allows the user to select a time (no time zone).

Depending on browser support, a time picker can show up in the input field.

Example

```
<form>
  <label for="appt">Select a time:</label>
  <input type="time" id="appt" name="appt">
</form>
```

Try it Yourself »

Input Type Url

The `<input type="url">` is used for input fields that should contain a URL address.

Depending on browser support, the url field can be automatically validated when submitted.

Some smartphones recognize the url type, and adds ".com" to the keyboard to match url input.

Example

```
<form>
  <label for="homepage">Add your homepage:</label>
  <input type="url" id="homepage" name="homepage">
</form>
```

Try it Yourself »

Input Type Week

The `<input type="week">` allows the user to select a week and year.

Depending on browser support, a date picker can show up in the input field.

Example

```
<form>
  <label for="week">Select a week:</label>
  <input type="week" id="week" name="week">
</form>
```

Try it Yourself »

HTML Exercises

Exercise:

In the form below, add an input field for text, with the name "username" .

```
<form action="/action_page.php">
<          >
</form>
```

Start the Exercise

HTML Input Type Attribute

Tag	Description
<input type="">	Specifies the input type to display

HTML Input Attributes

This chapter describes the different attributes for the HTML `<input>` element.

The value Attribute

The input `value` attribute specifies an initial value for an input field:

Example

Input fields with initial (default) values:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
</form>
```

Try it Yourself »

The readonly Attribute

The input `readonly` attribute specifies that an input field is read-only.

A read-only input field cannot be modified (however, a user can tab to it, highlight it, and copy the text from it).

The value of a read-only input field will be sent when submitting the form!

Example

A read-only input field:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John" readonly><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
</form>
```

Try it Yourself »

The disabled Attribute

The input `disabled` attribute specifies that an input field should be disabled.

A disabled input field is unusable and un-clickable.

The value of a disabled input field will not be sent when submitting the form!

Example

A disabled input field:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John" disabled><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
</form>
```

Try it Yourself »

The size Attribute

The input `size` attribute specifies the visible width, in characters, of an input field.

The default value for `size` is 20.

Note: The `size` attribute works with the following input types: text, search, tel, url, email, and password.

Example

Set a width for an input field:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" size="50"><br>
  <label for="pin">PIN:</label><br>
  <input type="text" id="pin" name="pin" size="4">
</form>
```

Try it Yourself »

The maxlength Attribute

The input `maxlength` attribute specifies the maximum number of characters allowed in an input field.

Note: When a `maxlength` is set, the input field will not accept more than the specified number of characters. However, this attribute does not provide any feedback. So, if you want to alert the user, you must write JavaScript code.

Example

Set a maximum length for an input field:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" size="50"><br>
  <label for="pin">PIN:</label><br>
  <input type="text" id="pin" name="pin" maxlength="4" size="4">
</form>
```

Try it Yourself »

The min and max Attributes

The input `min` and `max` attributes specify the minimum and maximum values for an input field.

The `min` and `max` attributes work with the following input types: number, range, date, datetime-local, month, time and week.

Tip: Use the max and min attributes together to create a range of legal values.

Example

Set a max date, a min date, and a range of legal values:

```
<form>
  <label for="datemax">Enter a date before 1980-01-01:</label>
  <input type="date" id="datemax" name="datemax" max="1979-12-
31"><br><br>

  <label for="datemin">Enter a date after 2000-01-01:</label>
  <input type="date" id="datemin" name="datemin" min="2000-01-
02"><br><br>

  <label for="quantity">Quantity (between 1 and 5):</label>
  <input type="number" id="quantity" name="quantity" min="1" max="5">
</form>
```

Try it Yourself »

The multiple Attribute

The input `multiple` attribute specifies that the user is allowed to enter more than one value in an input field.

The `multiple` attribute works with the following input types: email, and file.

Example

A file upload field that accepts multiple values:

```
<form>
  <label for="files">Select files:</label>
  <input type="file" id="files" name="files" multiple>
</form>
```

Try it Yourself »

The pattern Attribute

The input pattern attribute specifies a regular expression that the input field's value is checked against, when the form is submitted.

The pattern attribute works with the following input types: text, date, search, url, tel, email, and password.

Tip: Use the global title attribute to describe the pattern to help the user.

Tip: Learn more about regular expressions in our JavaScript tutorial.

Example

An input field that can contain only three letters (no numbers or special characters):

```
<form>
  <label for="country_code">Country code:</label>
  <input type="text" id="country_code" name="country_code"
  pattern="[A-Za-z]{3}" title="Three letter country code">
</form>
```

Try it Yourself »

The placeholder Attribute

The input placeholder attribute specifies a short hint that describes the expected value of an input field (a sample value or a short description of the expected format).

The short hint is displayed in the input field before the user enters a value.

The placeholder attribute works with the following input types: text, search, url, number, tel, email, and password.

Example

An input field with a placeholder text:

```
<form>
  <label for="phone">Enter a phone number:</label>
  <input type="tel" id="phone" name="phone"
  placeholder="123-45-678"
  pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">
</form>
```

Try it Yourself »

The required Attribute

The input `required` attribute specifies that an input field must be filled out before submitting the form.

The `required` attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

Example

A required input field:

```
<form>
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" required>
</form>
```

Try it Yourself »

The step Attribute

The input `step` attribute specifies the legal number intervals for an input field.

Example: if step="3", legal numbers could be -3, 0, 3, 6, etc.

Tip: This attribute can be used together with the max and min attributes to create a range of legal values.

The `step` attribute works with the following input types: number, range, date, datetime-local, month, time and week.

Example

An input field with a specified legal number intervals:

```
<form>
  <label for="points">Points:</label>
  <input type="number" id="points" name="points" step="3">
</form>
```

Try it Yourself »

Note: Input restrictions are not foolproof, and JavaScript provides many ways to add illegal input. To safely restrict input, it must also be checked by the receiver (the server)!

The autofocus Attribute

The input `autofocus` attribute specifies that an input field should automatically get focus when the page loads.

Example

Let the "First name" input field automatically get focus when the page loads:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" autofocus><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

Try it Yourself »

The height and width Attributes

The input `height` and `width` attributes specify the height and width of an `<input type="image">` element.

Tip: Always specify both the height and width attributes for images. If height and width are set, the space required for the image is reserved when the page is loaded. Without these attributes, the browser does not know the size of the image, and cannot reserve the appropriate space to it. The effect will be that the page layout will change during loading (while the images load).

Example

Define an image as the submit button, with height and width attributes:

```
<form>
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="image" src="img_submit.gif" alt="Submit" width="48" heig
ht="48">
</form>
```

Try it Yourself »

The list Attribute

The input `list` attribute refers to a `<datalist>` element that contains pre-defined options for an <input> element.

Example

An <input> element with pre-defined values in a <datalist>:

```
<form>
  <input list="browsers">
  <datalist id="browsers">
    <option value="Edge">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

Try it Yourself »

The autocomplete Attribute

The input `autocomplete` attribute specifies whether a form or an input field should have autocomplete on or off.

Autocomplete allows the browser to predict the value. When a user starts to type in a field, the browser should display options to fill in the field, based on earlier typed values.

The `autocomplete` attribute works with `<form>` and the following `<input>` types: text, search, url, tel, email, password, datepickers, range, and color.

Example

An HTML form with autocomplete on, and off for one input field:

```
<form action="/action_page.php" autocomplete="on">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email" autocomplete="off"><br><b
r>
  <input type="submit" value="Submit">
</form>
```

Try it Yourself »

Tip: In some browsers you may need to activate an autocomplete function for this to work (Look under "Preferences" in the browser's menu).

HTML Exercises

Exercise:

In the input field below, add placeholder that says "Your name here".

```
<form action="/action_page.php">
<input type="text" □ >
</form>
```

Start the Exercise

HTML Form and Input Elements

Tag	Description
<form>	Defines an HTML form for user input
<input>	Defines an input control

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Input form* Attributes

This chapter describes the different `form*` attributes for the HTML `<input>` element.

The form Attribute

The input `form` attribute specifies the form the `<input>` element belongs to.

The value of this attribute must be equal to the id attribute of the <form> element it belongs to.

Example

An input field located outside of the HTML form (but still a part of the form):

```
<form action="/action_page.php" id="form1">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <input type="submit" value="Submit">
```

```
</form>

<label for="lname">Last name:</label>
<input type="text" id="lname" name="lname" form="form1">
```

Try it Yourself »

The formaction Attribute

The input `formaction` attribute specifies the URL of the file that will process the input when the form is submitted.

Note: This attribute overrides the `action` attribute of the `<form>` element.

The `formaction` attribute works with the following input types: submit and image.

Example

An HTML form with two submit buttons, with different actions:

```
<form action="/action_page.php">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
  <input type="submit" formaction="/action_page2.php" value="Submit as
Admin">
</form>
```

Try it Yourself »

The formenctype Attribute

The input `formenctype` attribute specifies how the form-data should be encoded when submitted (only for forms with method="post").

Note: This attribute overrides the enctype attribute of the `<form>` element.

The `formenctype` attribute works with the following input types: submit and image.

Example

A form with two submit buttons. The first sends the form-data with default encoding, the second sends the form-data encoded as "multipart/form-data":

```
<form action="/action_page_binary.asp" method="post">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <input type="submit" value="Submit">
  <input type="submit" formenctype="multipart/form-data"
  value="Submit as Multipart/form-data">
</form>
```

Try it Yourself »

The formmethod Attribute

The input `formmethod` attribute defines the HTTP method for sending form-data to the action URL.

Note: This attribute overrides the method attribute of the `<form>` element.

The `formmethod` attribute works with the following input types: submit and image.

The form-data can be sent as URL variables (method="get") or as an HTTP post transaction (method="post").

Notes on the "get" method:

- This method appends the form-data to the URL in name/value pairs
- This method is useful for form submissions where a user want to bookmark the result
- There is a limit to how much data you can place in a URL (varies between browsers), therefore, you cannot be sure that all of the form-data will be correctly transferred
- Never use the "get" method to pass sensitive information! (password or other sensitive information will be visible in the browser's address bar)

Notes on the "post" method:

- This method sends the form-data as an HTTP post transaction
- Form submissions with the "post" method cannot be bookmarked
- The "post" method is more robust and secure than "get", and "post" does not have size limitations

Example

A form with two submit buttons. The first sends the form-data with method="get". The second sends the form-data with method="post":

```
<form action="/action_page.php" method="get">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit using GET">
  <input type="submit" formmethod="post" value="Submit using POST">
</form>
```

Try it Yourself »

The formtarget Attribute

The input `formtarget` attribute specifies a name or a keyword that indicates where to display the response that is received after submitting the form.

Note: This attribute overrides the target attribute of the `<form>` element.

The `formtarget` attribute works with the following input types: submit and image.

Example

A form with two submit buttons, with different target windows:

```
<form action="/action_page.php">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
  <input type="submit" formtarget="_blank" value="Submit to a new
window/tab">
</form>
```

Try it Yourself »

The formnovalidate Attribute

The input `formnovalidate` attribute specifies that an <input> element should not be validated when submitted.

Note: This attribute overrides the novalidate attribute of the `<form>` element.

The `formnovalidate` attribute works with the following input types: submit.

Example

A form with two submit buttons (with and without validation):

```
<form action="/action_page.php">
  <label for="email">Enter your email:</label>
  <input type="email" id="email" name="email"><br><br>
  <input type="submit" value="Submit">
  <input type="submit" formnovalidate="formnovalidate"
  value="Submit without validation">
</form>
```

Try it Yourself »

The novalidate Attribute

The `novalidate` attribute is a `<form>` attribute.

When present, novalidate specifies that all of the form-data should not be validated when submitted.

Example

Specify that no form-data should be validated on submit:

```
<form action="/action_page.php" novalidate>
  <label for="email">Enter your email:</label>
  <input type="email" id="email" name="email"><br><br>
  <input type="submit" value="Submit">
</form>
```

Try it Yourself »

HTML Form and Input Elements

Tag	Description
<form>	Defines an HTML form for user input
<input>	Defines an input control

For a complete list of all available HTML tags, visit our HTML Tag Reference.

HTML Canvas Graphics

The HTML `<canvas>` element is used to draw graphics on a web page.

The graphic to the left is created with `<canvas>`. It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.

What is HTML Canvas?

The HTML `<canvas>` element is used to draw graphics, on the fly, via JavaScript.

The `<canvas>` element is only a container for graphics. You must use JavaScript to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

Canvas is supported by all major browsers.

Canvas Examples

A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

The markup looks like this:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

Note: Always specify an `id` attribute (to be referred to in a script), and a `width` and `height` attribute to define the size of the canvas. To add a border, use the `style` attribute.

Here is an example of a basic, empty canvas:

Example

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid
#000000;">
</canvas>
```

Try it Yourself »

Add a JavaScript

After creating the rectangular canvas area, you must add a JavaScript to do the drawing.

Here are some examples:

Draw a Line

Example

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0, 0);
ctx.lineTo(200, 100);
ctx.stroke();
</script>
```

Try it Yourself »

Draw a Circle

Example

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95, 50, 40, 0, 2 * Math.PI);
ctx.stroke();
</script>
```

Try it Yourself »

Draw a Text

Example

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.font = "30px Arial";
ctx.fillText("Hello World", 10, 50);
</script>
```

Try it Yourself »

Stroke Text

Example

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
```

```
ctx.font = "30px Arial";
ctx.strokeText("Hello World", 10, 50);
</script>
```

Try it Yourself »

Draw Linear Gradient

Example

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");

// Create gradient
var grd = ctx.createLinearGradient(0, 0, 200, 0);
grd.addColorStop(0, "red");
grd.addColorStop(1, "white");

// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(10, 10, 150, 80);
</script>
```

Try it Yourself »

Draw Circular Gradient

Example

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");

// Create gradient
var grd = ctx.createRadialGradient(75, 50, 5, 90, 60, 100);
grd.addColorStop(0, "red");
grd.addColorStop(1, "white");

// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(10, 10, 150, 80);
</script>
```

Try it Yourself »

Draw Image

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
var img = document.getElementById("scream");
ctx.drawImage(img, 10, 10);
</script>
```

Try it Yourself »

HTML Canvas Tutorial

To learn more about `<canvas>`, please read our HTML Canvas Tutorial.