

2021

# Web Coding

# HTML & CSS

Learn the HTML and CSS that help designers build websites with a professional and consistent feel across the pages.





<input type="checkbox"/>	<b>1. Introduction to HTML</b>
<input type="checkbox"/>	<b>2. Inline Styles</b>
<input type="checkbox"/>	<b>3. Page Structure</b>
<input type="checkbox"/>	<b>4. Internal Style Sheets</b>
<input type="checkbox"/>	<b>5. Creating a Website</b>
<input type="checkbox"/>	<b>6. External Style Sheets</b>
<input type="checkbox"/>	<b>7. Divs</b>
<input type="checkbox"/>	<b>8. Images</b>
<input type="checkbox"/>	<b>9. Page Properties</b>
<input type="checkbox"/>	<b>10. Hyperlinks</b>
<input type="checkbox"/>	<b>11. Completing your Homepage</b>
<input type="checkbox"/>	<b>12. Tables</b>
<input type="checkbox"/>	<b>13. Background Images</b>
<input type="checkbox"/>	<b>14. HTML and CSS Summary</b>



HTML stands for *Hyper Text Markup Language*. This is the language that tells your browser how to construct a webpage so that it appears beautifully in your browser. The page itself is delivered as a simple text file. Images, videos and sounds are not part of the HTML but separate files called upon by the webpage. Styles such as fonts, colours and border designs are also delivered to you in a separate file. The browser puts all this information together and displays the completed webpage on your screen.



The HTML code is delivered to your browser. The HTML includes the text for the page and tells the browser where to find the images and styles needed to make the page appear as it was designed.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="description" value="My chess homepage" />
    <title>Chess Homepage</title>
    <link href="css/style.css" rel="stylesheet" type="text/css" />
  </head>
  <body>
    <div id="container">
      <div id="header">
        <a href="homepage.htm">img src="images/chess.jpg" width="100%" height="100%" />
      </div>
      <div id="sidebar">
        <h3>Years active</h3>
        <p>Spice. 5th century India to present</p>
        <h3>Genes</h3>
        <h3>Board game</h3>
        <p>Abstract strategy game</p>
        <h3>Player</h3>
      </div>
    </div>
  </body>
</html>
```



Images, videos and sounds are delivered separately to the browser.



Styles such as fonts, colours and border designs are usually delivered separately too.



HTML makes use of tags. Most HTML tags tell the web browser when to start doing something and then when to stop. For example, to produce the strong, emphasised text ***This is HTML*** on your page, you could use the following code:

```
<strong><em>This is HTML</em></strong>
```

The tags are the things in brackets. `<strong>` is the tag that instructs the browser to start using strong text. `</em>` is the tag that tells it to stop using emphasised text. The information below explains how the code works.

HTML Code	Tag name	Instruction to browser
<code>&lt;strong&gt;</code>	Strong	- start using strong, or important, type (it appears bold)
<code>&lt;em&gt;</code>	Emphasised	- start using emphasised type (it appears as italics)
This is HTML		- display the text
<code>&lt;/em&gt;</code>	Slash emphasised	- stop using emphasised type
<code>&lt;/strong&gt;</code>	Slash strong	- stop using strong type

**Jargon:** *HTML elements include start tags, content and end tags. In the code above, we have nested an emphasis element inside a strong element.*



## Task 1 – Typing HTML and Viewing it in a Browser

- a. Open a simple text editor on your computer. This may be Notepad (Windows), TextEdit (Mac) or any other simple text editing application. We are using Notepad++ as it colours the HTML code for us. The colours don't make any difference to how the code works; it's just easier to understand for the human.

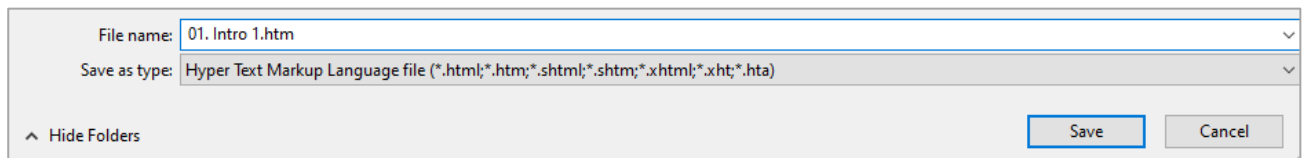
**Note:** Don't use Word Processing applications such as Microsoft Word as they hide too much secret code behind the scenes. Although Word suggests this can be removed, it's not always successful. You may use Web Design applications such as Dreamweaver, although the bells and whistles can be a distraction when coding; better to keep things simple. There are also lots of online HTML editors which allow you to type code and view the results in two halves of the screen (a search for 'online HTML editor' should do the trick) although these usually save the code to an online account. Notepad++ is really one of the best apps for these tasks.

- b. Type the text below, adding your own name in place of the word 'Rupert'.

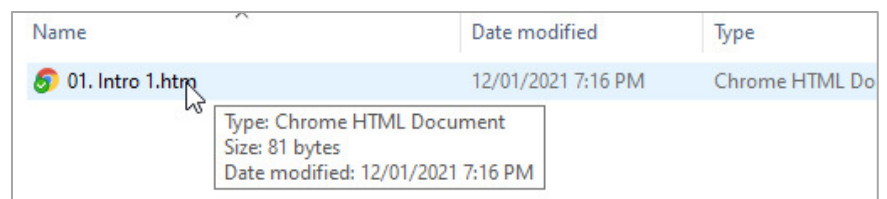
```
My name is <strong>Rupert</strong>
```

**Note:** In Wordpad++, select HTML from the Language menu to colour your code as above.

- c. Save the file as **01. Intro 1.htm**. If there is an option to do so in your application, select the file type 'HTML', 'htm' or 'Hyper Text Markup Language' from the list of choices. Remember the location of your saved file and leave your text editor open.

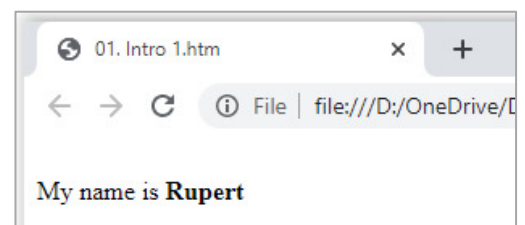


- d. Use the filing system on your computer to locate and open the file you have created.



- e. The file will open in whichever browser is set as the default on your computer (ours opens in Chrome). The text should appear something like ours on the right, with your name shown in bold (or strong) text.

Well done – you have created your first webpage. Isn't this easy?



- f. Return to your text editor and add the second line of text shown on the right. Save the file again.

```
My name is <strong>Rupert</strong>  
I am <em>100 years old</em>
```



## Search Engines

If you completed the previous activities, you should have created some styled text using lines of code like the one below.

```
<p style="color: blue; font-style: italic; font-weight: bold; font-size: 24px; font-family: Arial;">Some text</p>
```

The problem with declaring all the styles in this way is that it makes the code very messy. This is a problem for the poor web designer who's struggling to make sense of the HTML. It's also difficult for the search engines that are trying to work out what the page is actually about. Search engines send their robotic crawlers around looking for meaning in websites; they want to know what the content is related to so that they can send people to the right place when they search the web. However, the search engines' robots don't see a beautiful, formatted page; they must wade through all the code looking for the actual text. Wouldn't this be easier if there was just the code below?

```
<p class="mystyle">Some text</p>
```

That's much tidier. All we have to do now is tell the visitor that our style called *mystyle* should use the Arial font and be blue, bold, italic and 24 pixels high. This information can be stored well out of the way in something called a style sheet. The task of finding meaning will then be much more straightforward for the human and search engine alike.

## Efficiency

Have a look at the example HTML code below. There are three paragraphs all using blue, underlined, size 20 Verdana text.

```
<p style="color: blue; text-decoration: underline; font-size: 20px; font-family: Verdana;">Some text</p>
```

```
<p style="color: blue; text-decoration: underline; font-size: 20px; font-family: Verdana;">Some more text</p>
```

```
<p style="color: blue; text-decoration: underline; font-size: 20px; font-family: Verdana;">Even more text</p>
```

If you wanted to change the colour of all three paragraphs to red, how many places do we have to change the HTML code? Let's try something else. Look through the HTML below.

```
<Separate style code which states that the 'class' mystyle = blue, underlined, size 20, Verdana>
```

```
<p class="mystyle">Some text</p>
```

```
<p class="mystyle ">Some more text</p>
```

```
<p class="mystyle ">Even more text</p>
```

In how many places would you need to edit code to change the text to red now? Another advantage of separating styles is that we cut down the work required when changing the design of our website.

These are just two of the reasons why style sheets are preferred to writing inline styles.



## Task 1 – Creating an Internal Style Sheet

It's time to start separating the styles from the HTML. We are going to begin by placing our styles in style sheets at the top of the page, inside the head section of the code. Search engines know that they can ignore much of this section so they are better able to find meaning in the rest of the text.

- a. Make a copy of your template file from the *Page Structure* task and name the copy **04. Internal Style Sheets.htm**. Edit the code so that it is like that below. (If you didn't complete the earlier task, simply copy the code into a new file).

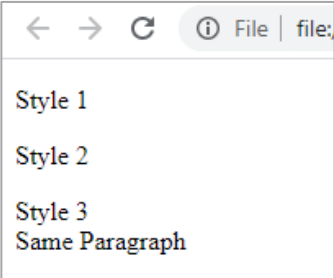
```
1 <!doctype html>
2 <html>
3
4 <head>
5 <meta charset="UTF-8">
6 <meta name="description" value="Using internal style sheets and CSS">
7 <title>Internal Style Sheets</title>
8 </head>
9
10 <body>
11 </body>
12
13 </html>
14
```

**Note:** We will start showing screenshots from our Notepad++ code so that we can refer to the line numbers where needed.

- b. Add three paragraphs between the `<body>` and `</body>` tags, as shown.

Check how the page appears in your browser.

```
10 <body>
11
12 <p>Style 1</p>
13 <p>Style 2</p>
14 <p>Style 3<br>
15     Same Paragraph
16 </p>
17
18 </body>
```

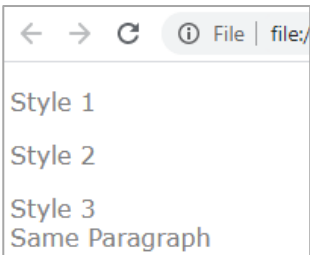


- c. Add style tags to the head section of your HTML (in the image on the right, ours open on line 9 and close on line 11). This will become your *style sheet*.

```
7 <title>Internal Style Sheet:</title>
8
9 <style type="text/css">
10
11 </style>
12
13 </head>
14
```

- d. We will start by setting out the styles for our paragraphs (notice the single 'p' used on line 11). These styles will be used in all paragraphs unless we tell the browser to use something different. Again, check the results in your browser.

```
9 <style type="text/css">
10     p {
11         color: gray;
12         font-family: Verdana;
13         font-size: 16px;
14     }
15
16
17 </style>
```



- e. Try varying the colours, fonts and sizes, noting that US spellings are used for colours. There is a list of colours you can name here: [https://www.w3schools.com/colors/colors\\_names.asp](https://www.w3schools.com/colors/colors_names.asp).





We will start the visual design of your website by adding some images to your homepage; one for the heading section and one lower down the page.

It is very common to find pictures on the internet to use on personal websites. Please remember that every image is owned by someone and you can't actually just take another person's property without their permission (especially if you are going to publish your website on the internet). Thankfully, there are a huge number of files on the web that you have permission to copy. Images from Wikimedia commons, Flickr and many other sources can often be legally used. An easy way to find these images is to use Google.

**Note:** *The images do not become part of the HTML; they remain as separate files which are 'collected' by the page as it is viewed.*

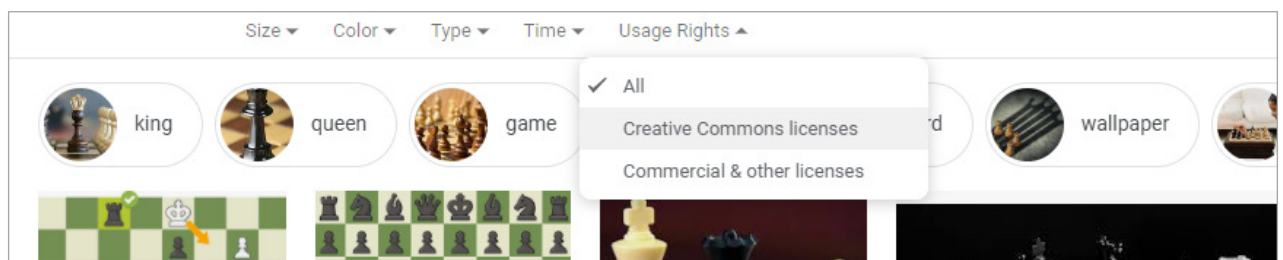
### Task 1 - Finding Legal Images

Our first goal is to find an image, or part of an image, that will make a good banner across the top of the page. The image should be short and wide.

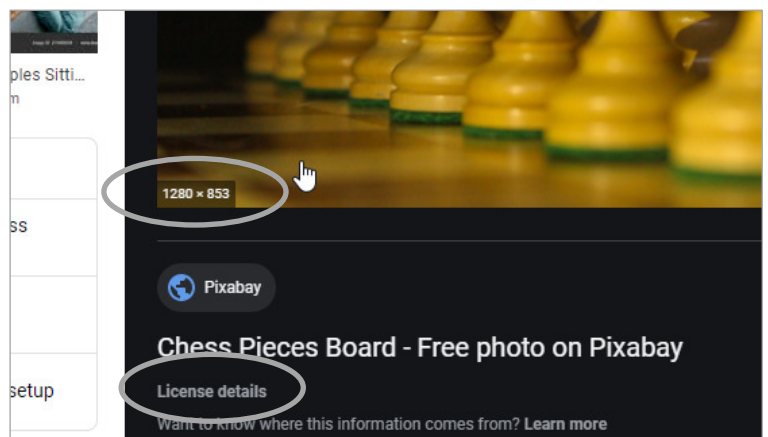
A possible target section of image.



- Perform a Google images search as usual. Click on *Tools* in the top menu and then on *Usage rights* in the submenu that appears. Select *Creative Commons licenses*.



- Click on an image you like so that it opens in the panel on the right.
- Mouse-over the image so that its dimensions are displayed in the lower left (as shown). Find one that is at least 960 pixels wide, or more if you are going to crop the picture. Remember that we are looking for a short, wide section of an image.





We are going to create and style a list of hyperlinks, linking to the other pages in our website. The links will be horizontally laid out across the *nav* div. Vertical lists are defined in the HTML but can then be set as horizontal lists in the CSS.

There are several HTML tags that help with the layout of lists.

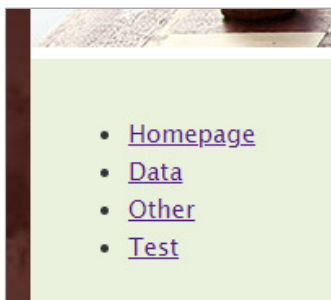
- `<ol> </ol>` create and close a numbered list (ordered list)
- `<ul> </ul>` create and close a bulleted list (unordered list)
- `<li> </li>` open and close each list item

### Task 1 - Creating a List in HTML

Add the HTML shown to your *nav* div in the homepage. Notice that the list opens and closes with the `<ul>` and `</ul>` tags and that each list item opens and closes with the `<li>` and `</li>` tags.

**Note:** If you backed up your earlier work, it will be much quicker to copy the list of links created previously. You can then change the `<p>` tags to `<li>`, the `</p>` tags to `</li>` and add the `<ul>` and `</ul>` tags.

The links will appear as a bulleted list for the moment. We will now go on to style them.



```
<div id="sidebar">Sidebar content</div>

<div id="nav">
  <ul>
    <li><a href="homepage.htm" title="Homepage">Homepage</a></li>
    <li><a href="data.htm" title="Data page">Data</a></li>
    <li><a href="other.htm" title="Other page">Other</a></li>
    <li><a href="test.htm" title="Test page">Test</a></li>
  </ul>
</div>

<div id="content">
```

### Task 2 - Styling Your Hyperlinks

Once the links are in place, we can add CSS styles to make them look prettier.

- a. Open your *style.css* file and add the following selector to the end. Save and reload the page in your browser. This should have the effect of making the list of links horizontal.

```
#nav li {
  display: inline;
}
```



**Note:** The 'li' code tells the browser to add the style to list elements; the '#nav' will cause the style to only be applied to lists in the 'nav' div. We could leave out the '#nav' but then the style would be applied to all lists.





### Task 2 - Styling Your Hyperlinks (cont.)

- b. Add some further code to this declaration block.

```
#nav li {  
    display: inline;  
    font-size: 26px;  
    list-style-type: none;  
}
```

- You may make any other font choices here.
- This tells the browser not to use bullets. It probably isn't anyway at this point.

- c. We will now add another declaration block, this time targeting the anchor tags (i.e. the hyperlinks) in the *nav* div. You should select your own colour.

```
#nav a {  
    background-color: #947AFF;  
}
```

- d. White text often looks good on a coloured background. The full hex value for the colour white is #FFFFFF, or we can use the shorthand #FFF. We also need some padding around the text, and things look better without the underlining (`text-decoration: none;`). Add the lines below to your declaration block then test some different values.

```
#nav a {  
    background-color: #947AFF;  
    color: #FFF;  
    padding: 10px 35px;  
    text-decoration: none;  
}
```



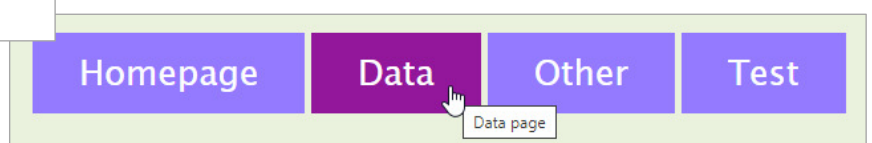
### Task 3 – Rollovers and other Link Effects

Rollover effects are seen when the mouse hovers over a link. You may change the colour and style of the text itself or change the background. You may also set styles for the link as it is clicked and after the page has been visited.

- |                        |   |
|------------------------|---|
| <code>a</code>         | <i>The initial hyperlink style</i>                        |
| <code>a:hover</code>   | <i>The style when the mouse is hovering over the link</i> |
| <code>a:active</code>  | <i>The style of the clicked link</i>                      |
| <code>a:visited</code> | <i>The style after the page has been visited</i>          |

Add some effects to your links such as ours below.

```
#nav a:hover {  
    background-color: #92179A;  
}
```





You should by now have enough skills to improve the quality of your homepage. You may do any of the following:

- Change or remove the remaining coloured backgrounds from your divs. Perhaps change the container div to white (#FFF) so that your text is on a white background where it can be read easily. You could use a colour dropper in a graphics application to select colours from your images.
- Improve the colours used for your navigation buttons, headings and text so that they fit with your header image colours.
- Add a margin to the left and right of your navigation buttons to separate them a little more.

- Add further images and text to the content and sidebar divs.

**Note:** We have used the heading style 'h3' in our sidebar with a negative value used for the bottom margin. This has the effect of pulling up the paragraph below. This can be a useful trick.


- Add links to other websites, perhaps in the footer. We have used the declaration `text-align: center;` to align this list of links in the centre of the div.
- Make any other changes that improve your page.

```
#container {
  background-color: #FFF;
  width: 960px;
  margin: auto;
}
```

```
#header {
```

```
h3 {
  font-size: 15px;
  margin: 30px 0px -15px 0px;
}
```

```
#footer {
  padding: 20px;
  background-color: #E6E0EC;
  clear: both;
  text-align: center;
}
#footer li {
  display: inline;
  font-size: 16px;
  list-style-type: none;
}
#footer a {
  padding: 0px 35px;
```




Homepage
Data
Choice
Test

### Chess

#### A Quick Overview

Chess is a two-player strategy board game played on a chessboard, a checkered gameboard with 64 squares arranged in an eight-by-eight grid. It is one of the world's most popular games, played by millions of people worldwide in homes, urban parks, clubs, online, correspondence, and in tournaments.

[Top Players](#)



**Years active**  
c. 6th-century India to present

**Genre**  
Board game  
Abstract strategy game

**Players**  
2

**Skills required**  
Strategy, tactics

Rules
Notation
History
Chess Computers