

HTML HyperText Markup language

Starting HTML

HyperText Markup Language (HTML) is the main markup language for displaying web pages and other information that can be displayed in a web browser. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like **<html>**), within the web page content.

HTML tags most commonly come in pairs like **<h1>** and **</h1>**, although some tags, known as empty elements, are unpaired, for example ****. The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, tags, comments and other types of text-based content.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts in languages such as JavaScript which affect the behavior of HTML webpages.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicitly presentational HTML markup.

Writing HTML Code Part 1

To create basic code for an HTML page you can use a simple text editor and save the file with **.html** after the file name such as **webpage.html** you can also use some free modern, open source text editors that understand web design you can find links to these editors in the Links section.

the basic code used for creating a HTML page is as follows:
At the top of the page is what is known as the document type or DTD further reading and explanation of DTD can be found at www.w3.org

Here we are using the HTML5 document type.

<!DOCTYPE html>

Below that is the starting HTML tag **<html lang="en">** the lang="en" indicates what language the page is written in, the en indicating that English is the language used on the page.

The head section **<head></head>** of the document is used to provide the search engines and the various browsers information about a given Web page, they help search engines categorize the web page correctly.

between the head tags the title section gives the page it's title in the case of the example Building WebSites, this will appear at the top of the browser **<title>Building WebSites</title>**

The head section also includes the meta elements beginning with the content type this specifies which character encoding is being used on the page, For HTML the default character encoding is UTF-8. **<meta charset=UTF-8>**

The viewport meta tag instructs the browser to use the actual device width making this compatible for mobile phones **<meta name="viewport" content="width=device-width, initial-scale=1">**

Next is a description of the pages content **<meta name="description" content="building web sites provides a free html and css tutorial of how to build a web site">**

The keywords meta tag used to input keywords that can be found on the page such as headers and the keywords of the pages content **<meta name="keywords" content="building web sites, web design, html tutorial, css tutorial, building a web site.">**

And finally in the head section of the document is a link to an external cascading style sheet which will give more control over the web site as it is being developed making it easier to make changes from one place **<link rel="stylesheet" type="text/css" href="style.css">**.

The code for the Webpage should look like the example below

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Building WebSites</title>
    <meta charset=UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <meta name="description" content="building web sites
provides a free html and css tutorial of how to build a web
site">
    <meta name="keywords" content="building web sites, web
design,html tutorial, css tutorial, building a web site.">
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <h1>How to Build a Website</h1>

    <p>This page was created for my studies on HTML, CSS and
JavaScript,the aim is to build a website learning and using the
various code to construct a website that can be viewed on all
platforms ie. Mobile phones and tablets as well as desktop
computers and laptops.</p>

  </body>
</html>
```

A useful tool when you've finished writing your first page is to check if your code validates with the W3C standards using the W3C validator [w3 code validator](http://validator.w3.org)

w3 code validator

<http://validator.w3.org>

Writing HTML Code Part 2

Content

When thinking of the type of content the web site would include we need to consider the style and layout of the pages, you should know what is the most

important information that your visitors need to know about your web site and how best to visually present it, In previous versions of HTML creating new sections of content was done almost exclusively through the use of heading tags, with **<div></div>** tags being used to group areas of content together. Class or ID attributes would allow designers to attach more semantic meaning to the groups.

In HTML5 new sectioning and semantic elements were introduced to help represent commonly used sections and create more consistent page structures. Tags like the **aside**, **article**, **footer**, **header**, **main**, **nav**, and **section** allow you to identify common page regions. Below are some examples of the other things that can be used to help you with your visual presentation.

Headings

To create headings on a web page the **<h1>**Insert your headings here **</h1>** start and end tags are used, there are six sizes h1, h2, h3, h4, h5 and h6 used in HTML5.

Paragraphs

Using the Paragraph and the Single Line Break tag, paragraphs are created using the **<p>**Insert your text here**</p>** start and end tags, also used in this example is the **
** single line break tag.

Other Tags used

To emphasize text you can use the **** tags, This text is bold or the **** tags to italicize the text This text is italicized you can also underline text using the **<ins></ins>** tags, other tags are the **** which elevates the text and the **** tags which lowers the text if you wish to show that text is no longer valid you can use the **** tag such as this is no longer valid

The code examples on this page use the **<pre></pre>** tags these tags allow you to move the text around to resemble how code would be written such as

```
<ul class="a">
  <li>This is number one</li>
  <li>This is number two</li>
  <li>This is number three</li>
</ul>
```

Creating an Horizontal Rule

Horizontal Rules can be placed to create divisions on a web site, the following code is used to create these **<hr>**

Using the img tag to insert Images

Inserting graphics to the web page is done by using the **** attribute. Inserting the name of the file with its extension in quotation marks and using the **<alt="">** attribute to give a description of the graphics file as in the example ****



Creating Links

If you want to create links to other web sites on your own web site use the anchor tag, you need to make sure you use the full URL address of the site you're linking to and if you use **target="_blank"** this will open the site in another page so that your visitors can return to your web site after using the link that you have referenced, as in the example below

<a href="http://www.w3schools.com
target="_blank">www.w3schools.com this takes you to w3schools's homepage.

www.w3schools.com

Creating links to other pages on your web site using **Links Page** this takes you to a web sites Links page.

Creating links to other locations on the same page on your web site using the **Back to the top of the page** and the **<h1 id="Top">Building WebSites</h1>** heading takes you to the top of a web page.

Writing HTML Code Part 3

Creating ordered and unordered lists

Creating ordered and unordered lists using the ****, **** and **** tags, there are a number of different attributes that can be added using the cascading style sheet **list-style-type:decimal;** which is the default, upper and lowercase letters **list-style-type:upper-alpha;** or **list-style-type:lower-alpha;** Roman numerals using the upper and lowercase **list-style-type:upper-roman;** and **list-style-type:lower-roman;** and in a similar fashion for the unordered list the **list-style-type:square;** **list-style-type:circle;** and **list-style-type:disc;**

Below is the HTML code used to create an unordered list

```
<ul class="a">
  <li>This is number one</li>
  <li>This is number two</li>
  <li>This is number three</li>
</ul>
```

The CSS code used to style the unordered list which creates a disc shaped bulleted list

```
.a {list-style-type:disc;}
```

Below is the HTML code used to create an ordered list

```
<ol class="e">
  <li>This is number one</li>
  <li>This is number two</li>
  <li>This is number three</li>
</ol>
```

The CSS code used to style the ordered list which creates a decimal list

```
.a {list-style-type:decimal;}
```

Creating Tables

Tables can be useful with the layout of certain types of content, to form a table you basically need three HTML elements: **<table>**, **<tr>** (Table Row) and **<td>** (Table Data) below is the HTML and CSS code used to create a table.

```

<table>
  <tr>
    <td>Selector</td>
    <td>Declaration</td>
  </tr>
  <tr>
    <td>h1</td>
    <td>{ color= #0000ff;}</td>
  </tr>
  <tr>
    <td>h1</td>
    <td>{ font-size= 2em;}</td>
  </tr>
  <tr>
    <td></td>
    <td>Property Value</td>
  </tr>
</table>

```

The CSS code used to style the table

```

#table1 {
  border:1px solid #000;
  margin:auto;
  width:400px;
}

table tr td {
  width:400px;
  height:auto;
  padding:5px;
  border:1px solid #000;
}

```

The HTML and CSS code would produce the following table

Selector	Declaration
h1	{ color: #0000ff;}
h1	{ font-size: 2em;}
	Property Value

Contact Details

The code used to create the contact form

```
<form class="center" action="mailto:info@building-  
websites.org"  
  method="post" enctype="text/plain">  
  <label>Your Name</label>  
  <input type="text" name="yourname" value="">  
  <label>Your e-mail</label>  
  <input type="email" name="youremail" value="">  
  <label>Your Message</label>  
  <textarea name="yourmessage"  
rows="10" cols="30"></textarea>  
  <input type="submit" name="submit" value="submit">  
</form>
```



Embedding Video and Music Files

The HTML code for embedding the Video

```
<div class="center">  
  <video controls class="video">  
    <source src="video/fireworks.mp4"  
type="video/mp4">  
    Your browser does not support the video tag.  
  </video>  
</div>
```

The CSS code for the Video

```
.video {  
  margin:auto;  
  width:40%;  
}
```

The HTML code for embedding a music file

```
<div class="center">  
  <audio controls class="audio">  
    <source src="music/Keren.ogg" type="audio/ogg">  
    <source src="music/Keren.mp3"  
type="audio/mpeg">  
    Your browser does not support the audio element.  
  </audio>  
</div>
```

The CSS code for the Audio

```
.audio {  
  border-style:none;  
  margin:auto;  
  width:20%;  
  box-shadow: 5px 5px 20px rgba(0,0, 0,  
0.4); border-radius: 90px;  
  transform: scale(1.05);  
}
```

Creating a Slideshow

The image files were cropped using an image editor to make them all the same height and width.

HTML code for the slideshow

```
<div class="Slides">  
  
```

```



</div>
```

CSS code for the slideshow

```
.Slides {
    max-width: 60%;
    height:20%;
    margin: auto;
    border:1px solid
    #000; border-
    radius:1em;
```

Javascript code for the slideshow

```
<script>
var slideIndex = 0;
carousel();

function carousel() {
    var i;
    var x = document.getElementsByClassName("mySlides");
    for (i = 0; i < x.length; i++) {
        x[i].style.display = "none";
    }
    slideIndex++;
    if (slideIndex > x.length) {
        slideIndex = 1
    }
    x[slideIndex - 1].style.display = "block";
    setTimeout(carousel, 2000); // Change image every 2
seconds
}
</script>
```