**Learn HTML, CSS & 960-Grid in 60 minutes**

- by Nikhil Patni

**HTML Basics**

Welcome to HTML Basics. This workshop leads you through the basics of Hyper Text Markup Language (HTML). HTML is the building block for web pages. You will learn to use HTML to author an HTML page to display in a web browser.

**Objectives:**

By the end of this workshop, you will be able to:

* Use a text editor to author an HTML document.
* Be able to use basic tags to denote paragraphs, emphasis or special type.
* Create hyperlinks to other documents.
* Create an email link.
* Add images to your document.
* Use a table for layout.
* Apply colors to your HTML document.

**Prerequisites:**

You will need a text editor, such as Notepad++ (<https://notepad-plus-plus.org/>) and a web browser. Any web browser will do just fine.

**Mac Users:** SimpleText is the default text editor on the Mac. In OSX use TextEdit and changethe following preferences: Select (in the preferences window) Plain text instead of Rich text and then select Ignore rich text commands in HTML files. This is very important because if you don't do this HTML codes probably won't work.

One thing you should avoid using is a word processor (like Microsoft Word) for authoring your HTML documents.

**What is an html File?**

HTML is a format that tells a computer how to display a web page. The documents themselves are plain text files with special "tags" or codes that a web browser uses to interpret and display information on your computer screen.

* HTML stands for Hyper Text Markup Language
* An HTML file is a text file containing small markup tags
* The markup tags tell the Web browser how to display the page
* An HTML file must have an htm or html file extension

**Try It?**

Open your text editor and type the following text:

<!DOCTYPE html>

<html>

<head>

<title>My First Webpage</title>

</head>

<body>

This is my first homepage. <b>This text is bold</b>

</body>

</html>

Save the file as **index.html**. Now move to the folder where you stored this file and double click the file to open it in your default browser. You can also select other browser by right clicking and select **open with**.

**Example Explained**

What you just made is a skeleton html document. This is the minimum required information for a web document and all web documents should contain these basic components. The first tag in your html document is <!DOCTYPE html> which tells your browser that this is a document type and this convention denotes the latest version of HTML which is HTML5. The next tag in your html document is <html>. This tag tells your browser that this is the start of an html document. The last tag in your document is </html>. This tag tells your browser that this is the end of the html document.

The text between the <head> tag and the </head> tag is header information. Header information is not displayed in the browser window.

The text between the <title> tags is the title of your document. The <title> tag is used to uniquely identify each document and is also displayed in the title bar of the browser window.

The text between the <body> tags is the text that will be displayed in your browser.

The text between the <b> and </b> tags will be displayed in a bold font.

**HTM or HTML Extension?**

When you save an HTML file, you can use either the .htm or the .html extension. The .htm extension comes from the past when some of the commonly used software only allowed three letter extensions. It is perfectly safe to use either .html or .htm, but be consistent. **index.htm** and index.html are treated as different files by the browser.

**HTML Elements**

Remember the HTML example from the previous page:

<html>

<head>

<title>My First Webpage</title>

</head>

<body>

This is my first homepage. <b>This text is bold</b>

</body>

</html>

This is an HTML element:

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

The HTML element begins with a start tag: <b>

The content of the HTML element is: This text is bold

The HTML element ends with an end tag: </b>

The purpose of the <b> tag is to define an HTML element that should be displayed as bold.

This is also an HTML element:

<body>

This is my first homepage. <b>**This text is bold**</b> </body>

This HTML element starts with the start tag <body>, and ends with the end tag </body>. The purpose of the <body> tag is to define the HTML element that contains the body of the HTML document.

**Note:** It doesn't matter which tag is first, but they must be closed in the proper order.

**Basic HTML Tags**

The most important tags in HTML are tags that define headings, paragraphs and line breaks.

**<html>** - Defines an HTML document

**<body>** - Defines the document's body

**<h1> to <h6>** - Defines header 1 to header 6

**<p>** - Defines a paragraph

**<br>** - Inserts a single line break

**<hr>** - Defines a horizontal rule

**<!-- -->** - Defines a comment

**Headings**

Headings are defined with the <h1> to <h6> tags. <h1> defines the largest heading while <h6> defines the smallest.

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

**<h2>This is a heading</h2>**

**<h3>This is a heading</h3>**

**<h4>This is a heading</h4>**

**<h5>This is a heading</h5>**

**<h6> This is a heading</h6>**

HTML automatically adds an extra blank line before and after a heading. A useful heading attribute is align.

<h5 align="left">I can align headings </h5>

<h5 align="center">This is a centered heading </h5>

<h5 align="right">This is a heading aligned to the right </h5>

**Paragraphs**

Paragraphs are defined with the <p> tag. Think of a paragraph as a block of text. You can use the align attribute with a paragraph tag as well.

<p align="left">This is a paragraph</p>

<p align="center">this is another paragraph</p>

**Line Breaks**

The <br> tag is used when you want to start a new line, but don't want to start a new paragraph.

**Comments in HTML**

The comment tag is used to insert a comment in the HTML source code.

<!-- This is a Comment -->

**Other HTML Tags**

Visit <https://www.quackit.com/html/tags/> for complete list of HTML5 tags.

Character tags like <strong> and <em> produce the same physical display as <b> and <i> but are more uniformly supported across different browsers.

**HTML Backgrounds**

**Backgrounds**

The <body> tag has two attributes where you can specify backgrounds. The background can be a color or an image.

**Bgcolor**

The bgcolor attribute specifies a background-color for an HTML page. The value of this attribute can be a hexadecimal number, an RGB value, or a color name:

<body bgcolor="#000000">

<body bgcolor="rgb(0,0,0)">

<body bgcolor="black">

The lines above all set the background-color to black.

**Background**

The background attribute can also specify a background-image for an HTML page. The value of this attribute is the URL of the image you want to use. If the image is smaller than the browser window, the image will repeat itself until it fills the entire browser window.

<body background="clouds.gif">

**Try It Out!**

Open your text editor and type the following text

<html>

<head>

<title>My First Webpage</title>

</head>

<body background=" https://static.pexels.com/photos/110854/pexels-photo-110854.jpeg " bgcolor="#EDDD9E">

<h1 align="center">My First Webpage</h1>

<p>Welcome to my <strong>first</strong> webpage. I am writing this page using a text editor and plain old html.</p>

<p>By learning html, I'll be able to create webpages like a <del>beginner</del> pro....<br>

which I am of course.</p>

</body>

</html>

Save your page as **mypage3.html** and view it in your browser.

Notice we gave our page a background color as well as a background image. If for some reason the web page is unable to find the picture, it will display our background color.

**HTML Lists**

HTML provides a simple way to show unordered lists (bullet lists) or ordered lists (numbered lists).

**Unordered Lists**

An unordered list is a list of items marked with bullets (typically small black circles). An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

|  |  |  |
| --- | --- | --- |
| **This Code** |  | **Would Display** |
| <ul> |  |  |
| <li>Coffee</li> |  | * Coffee |
| <li>Milk</li> |  | * Milk |
| </ul> |  |  |

**Ordered Lists**

An ordered list is also a list of items. The list items are marked with numbers. An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

|  |  |  |
| --- | --- | --- |
| **This Code** |  | **Would Display** |
| <ol> |  |  |
| <li>Coffee</li> |  | 1. Coffee |
| <li>Milk</li> |  | 1. Milk |
| </ol> |  |  |

Inside a list item you can put paragraphs, line breaks, images, links, other lists, etc.

**Definition Lists**

Definition lists consist of two parts: a **term** and a **description**. To mark up a definition list, you need three HTML elements; a container <dl>, a definition term <dt>, and a definition description <dd>.

|  |  |
| --- | --- |
| **This Code** | **Would Display** |
| <dl> |  |
| <dt>Cascading Style Sheets</dt> | Cascading Style Sheets |
| <dd>Style sheets are used to provide | Style sheets are used to provide |
| presentational suggestions for | presentational suggestions for |
| documents marked up in HTML | documents marked up in HTML. |
| </dd> |
|  |
| </dl> |  |

Inside a definition-list definition (the <dd> tag) you can put paragraphs, line breaks, images, links, other lists, etc.

**Try It Out**

Open your text editor and type the following:

<html>

<head>

<title>My First Webpage</title>

</head>

<body bgcolor="#EDDD9E">

<h1 align="center">My First Webpage</h1>

<p>Welcome to my <strong>first</strong> webpage. I am writing this page using a text editor and plain old html.</p>

<p>By learning html, I'll be able to create web pages like a pro....<br>

which I am of course.</p>

Here's what I've learned:

<ul>

<li>How to use HTML tags</li>

<li>How to use HTML colors</li>

<li>How to create Lists</li>

</ul>

</body>

</html>

Save your page as **mypage4.html** and view it in your browser.

**HTML Styles – CSS**

**CSS stands for Cascading Style Sheets.**

CSS describes how HTML elements are to be displayed on screen, paper, or in other media.

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

CSS can be added to HTML elements in 3 ways:

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

The most common way to add CSS, is to keep the styles in separate CSS files. However, here we will use inline and internal styling, 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.

This example sets the text color of the <h1> element to blue:

<!DOCTYPE html>

<html>

<body>

<h1 style="color:blue;">This is a Blue Heading</h1>

</body>

</html>

Output :-

# This is a Blue Heading

## 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:

<!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>

Output :-

# This is a heading

This is a paragraph

## **External CSS**

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

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

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

<!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>

An 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 how the "styles.css" looks:

body {  
 background-color: powderblue;  
}  
h1 {

color: blue;  
}  
p {

color: red;  
}

Output :-

# This is a heading

This is a paragraph.

## CSS Fonts

The CSS color property defines the text color 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.

<!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>

## CSS Border

The CSS border property defines a border around an HTML element:

p {

border: 1px solid powderblue;  
}

**CSS Padding**

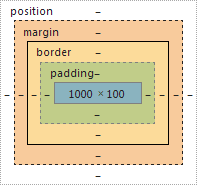
The CSS padding property defines a padding (space) between the text and the border:

p {  
 border: 1px solid powderblue;  
 padding: 30px;  
}

**CSS Margin**

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

p {  
 border: 1px solid powderblue;  
 margin: 50px;  
}



**The id Attribute**

To define a specific style for one special element, add an id attribute to the element:

<p id="p01">I am different</p>

then define a style for the element with the specific id:

#p01 {  
 color: blue;  
}

Note: The id of an element should be unique within a page, so the id selector is used to select one unique element!

**The class Attribute**

To define a style for a special type of elements, add a class attribute to the element:

<p class="error">I am different</p>

then define a style for the elements with the specific class:

p.error {  
 color: red;  
}

**External References**

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

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

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

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

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

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

For further CSS3 Quick reference sheet visit :-

<https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/d7fb67af-5180-463d-b58a-bfd4a220d5d0/css3-cheat-sheet.pdf>

**HTML Links**

HTML uses the <a> anchor tag to create a link to another document or web page.

<a href="url">Text to be displayed</a>

**The Target Attribute**

With the target attribute, you can define **where** the linked document will be opened. By default, the link will open in the current window. The code below will open the document in a new browser window:

<a href="<http://www.thapar.edu/>" target=”\_blank”>Visit Thapar Website!</a>

**Email Links**

To create an email link, you will use mailto: plus your email address.

<a href="<mailto:npatni@sapient.com>">Personal Email id</a>

To add a subject for the email message, you would add ?subject= after the email address. For example:

<a href="mailto:[npatni@sapient.com](mailto:npatni@sapient.com)?subject=Personal Assistance">Personal Email id</a>

**HTML Images**

**The Image Tag and the Src Attribute**

<img src=”css/image.jpg”> - This tag add an image to html webpage and pick that image from source folder named css.

The alt attribute is used to define an alternate text for an image.

<img src="css/image.jpg" alt="Alternating text">

**Tables**

<table title=” ” >

<tr>

<th></th>

<th></th>

<th></th>

</tr>

<tr>

<td></td>

<td></td>

<td></td>

</tr>

</table>

**<table></table> tags are used for creating tables.**

**<tr></tr> tags used for creating rows./slideshows/writing-css/#/10**

**<th></th> tags are used for column headings.**

**<td></td> tags are for table data.**

**HTML Block and Inline Elements:**

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

The <div> element is a block-level element.

<div>Hello</div>  
<div>World</div>

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> <output> <p>

<pre> <section> <table> <tfoot>

<ul> <video>

## **The <div> Element**

The <div> element is often used as a container for other HTML elements.

The <div> element has no required attributes, but both style and class are common.

When used together with CSS, the <div> element can be used to style blocks of content:

<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>

## **The <span> Element**

The <span> element is often used as a container for some text.

The <span> element has no required attributes, but both style and class are common.

When used together with CSS, the <span> element can be used to style parts of the text:

<h1>My <span style="color:red">Important</span> Heading</h1>

**HTML Iframe**

An iframe is used to display a webpage within a webpage.

<iframe src="URL"></iframe>

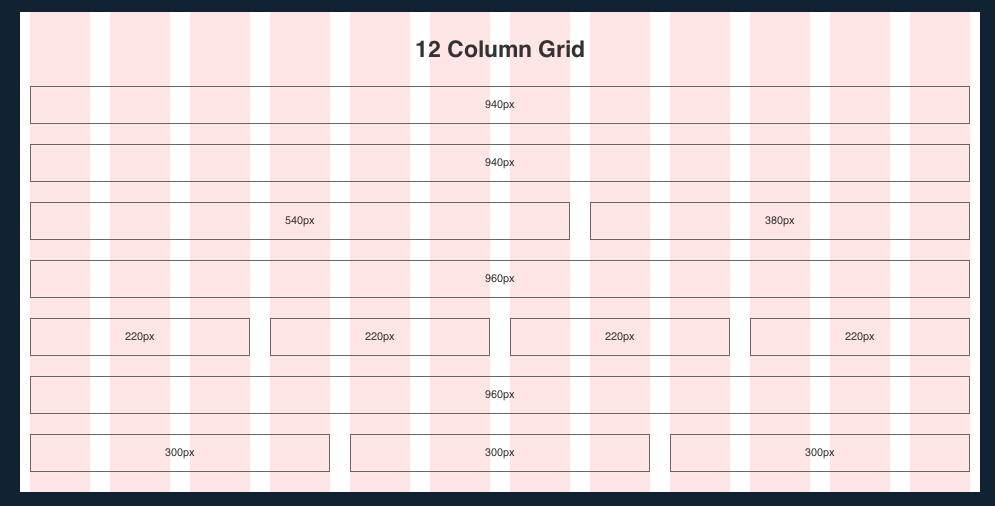
<iframe src="demo\_iframe.htm" height="200" width="300"></iframe>

You can also set width and height to an iframe.

**960 Grid System**

The 960 grid system is an effort to streamline web development by providing commonly used dimensions, based on a width of 960 pixels.

There are two variants: 12 and 16 columns, which can be used separately or in tandem. If you like, there is also a 24-column version. The 12-column grid is divided into portions that are 60 pixels wide. The 16-column grid consists of 40 pixel increments. Each column has 10 pixels of margin on the left and right, which create 20 pixel wide gutters between columns.



The 24-column grid is also included. It consists of columns 30 pixels wide, with 10 pixel gutters, and a 5 pixel buffer on each side of the container.

Ehhh... Show me teh codez!!!

**Using the 960 Grid System**

The 960 Grid is almost frighteningly easy to use (in fact, you may start to ask yourself: "Why wasn't I using this earlier?"), with only a few simple rules you need to abide by.

#### [Download the 960 Grid from 960.gs](http://960.gs/)

You can find a lot more information about using and prototyping with CSS frameworks at [Nettuts](http://net.tutsplus.com/articles/news/a-detailed-look-at-the-960-css-framework/?_ga=2.139829491.1388362256.1517850477-867684778.1517850477), so I'm just going to cover the basics here.

First, you'll need to reference the appropriate CSS files into your design: reset.css (the reset file that accounts for browser quirks), text.css (the built-in typography stylesheet), and 960.css (or 960\_24\_col.css if you are using a 24 columns grid). Usually, I place all of my site-specific styling in a separate file (usually called style.css), to ensure I don't break anything in the Framework.

Here's a quick overview of the syntax, pulled from the 960.gs site:

<div class="container\_12">

<div class="grid\_7 prefix\_1">

<div class="grid\_2 alpha">

...

</div>

<div class="grid\_3">

...

</div>

<div class="grid\_2 omega">

...

</div>

</div>

<div class="grid\_3 suffix\_1">

...

</div>

</div>

The first div is your container, what your content will be stored in. By defining it's class as "container\_12", you're saying: "I want this container to have a 12 column grid" (The same practice applies when using a 16 and 24 column grid).

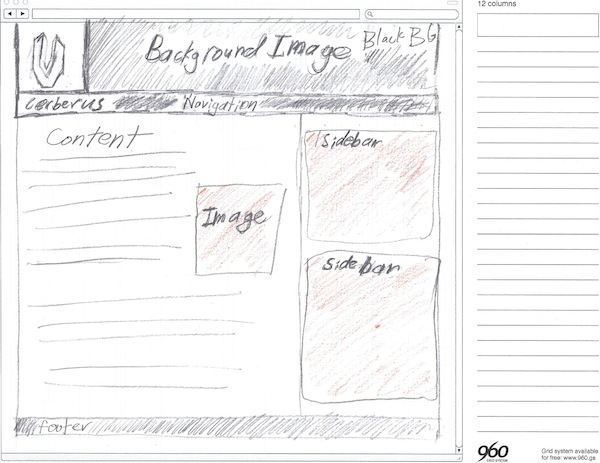
Now that you've defined the container as a 12 column layout, you begin to structure your site in such a way that it fits within the layout. If you look at the next div elements in the hierarchy (as noted by the tabs), you'll see the syntax for their definitions. "Grid\_7" and "grid\_3" are easy enough to understand: you're saying that this div element is going to take up X number of columns. However, 7+3 = 10 < 12, which means you'll have 2 empty columns at the end of the container, right? Well, this is where the prefix and suffix aspect of the class definition come into play. With a class definition of "grid\_x suffix\_y", you're saying: "I want this div to take up X columns, and for there to be Y number of empty columns after it." (This also applies to prefix). So, since "Grid\_7" has a prefix of 1, and "Grid\_3" has a suffix of 1, 7+3+1+1 = 12!

By now, I know you're wondering about the Div elements nested in the "Grid\_7" div. Well, you can nestle divs which take up a specified number of columns inside another div, so long as it does not exceed the size of its parent grid. Furthermore, the first and last elements must have "alpha" and "omega" appended to their class definition, respectively. So, in our sample code: "grid\_2 alpha" is the first div nested within "grid\_7", the next element "grid\_3" is in the middle (and therefore does not need an "alpha" or "omega" appended to it), and finally "grid\_2\_ omega" is the last div nested within "grid\_7". And since 2+3+2 = 7, the definition is proper.

Let’s have a look at another example :-

|  |
| --- |
| <div id="container" class="container\_12">  <div id="header" class="grid\_12">  <div id="logo" class="grid\_2 alpha">Logo</div>  <div id="subHeader" class="grid\_12 omega">  <div id="tagline" class="grid\_4 alpha"><h1>Cerberus: Strength for Humanity</h1></div>  <div class="grid\_8 omega">Navigation</div>  </div>  </div>  <div id="content" class="grid\_8">  Content  </div>  <div id="sidebar" class="grid\_4">  <div class="sidebarSection">  Sidebar  </div>  <div class="sidebarSection">  Sidebar  </div>  <div class="sidebarSection">  Sidebar  </div>  </div>  <div id="footer" class="grid\_12">Footer</div>  </div> |

This code will give output like this :-



Hopefully you understand basics of HTML, CSS and 960-Grid framework through this document. Now you are ready to fly on your own. Go to something called “[www.google.com](http://www.google.com/)” and explore more on these topics.

Until next time – Sayonara :)