Basics of HTML & CSS

Ranveer Aggarwal
Department of Computer Science and Engineering
IIT Bombay

Introduction

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:

- Author an HTML document.
- Be able to use basic html tags to denote various stuff.
- Create hyperlinks to other documents.
- Create an email link.
- Add images to your document.
- Use a table for layout.
- Add colors to your HTML document.

Prerequisites:

- A text editor like Notepad, Notepad++, Sublime Text etc. with Sublime Text being the most preferable.
- A web browser like Google Chrome, Mozilla Firefox, Internet Explorer etc.

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

Do It Yourself

Open your text editor and type the following text:

```
<html>
        <head>
                <title>My First Page</title>
        </head>
        <body>
                This is my first homepage.
                <b>This text is bold</b>
        </body>
</html>
```

Now you would like to see how the browser interprets your code. So, just save the file as mypage.html. Now go to the file directory and right-click on the file and click on Open With and then click on Google Chrome (or any other web browser).

To view how the page should look, visit this web page:
`www.cse.iitb.ac.in/~ranveer/tutorials/html/mypage.html`

Explanation of the Example

You just made 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 <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.

Extensions

When you save an HTML file, you can use either the .htm or the .html extension. The .htm extension was used earlier 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. page.htm and page.html are treated as different files by the browser.

View HTML Source

"Stealing" is a good way to learn HTML. To see other peoples' code, simply click on the View option in your browsers toolbar and select Source or Page Source. This will open a window that shows you the actual HTML of the page. Go ahead and view the source html for this page.

HTML Tags

What are HTML tags?

- HTML tags are used to mark-up HTML elements
- HTML tags are surrounded by the two characters < and >
- The surrounding characters are called angle brackets
- HTML tags normally come in pairs like <b> and </b>
- The first tag in a pair is the start tag, the second tag is the end tag
- The text between the start and end tags is the element content
- HTML tags are not case sensitive, <b> means the same as <B>

Logical v/s Physical Tags

In HTML there are both logical tags and physical tags. Logical tags are designed to describe (to the browser) the enclosed text's meaning. An example of a logical tag is the <strong> </strong> tag. By placing text in between these tags you are telling the browser that the text has some greater importance. By default all browsers make the text appear bold when in between the <strong> and </strong> tags.

Physical tags on the other hand provide specific instructions on how to display the text they enclose.
Examples of physical tags include:

- <b>: Makes the text bold.

- <big>: Makes the text usually one size bigger than what's around it.
- <i>: Makes text italic.

Physical tags were invented to add style to HTML pages because style sheets were not around, though the original intention of HTML was to not have physical tags. Rather than use physical tags to style your HTML pages, you should use style sheets (which will be introduced later).

HTML Elements

Remember the HTML example from the previous section:

```
<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.

Nested Tags

You may have noticed in the example above, the <body> tag also contains other tags, like the <b> tab. When you enclose an element in with multiple tags, the last tag opened should be the first tag closed.
For example:

```
<p> <b> <em>This is NOT the proper way to close nested
tags.</p> </em> </b>

<p> <b> <em>This is the proper way to close nested
tags. </em> </b> </p>
```

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

Why Use Lowercase Tags?

You may notice lowercase tags have been used even though it was earlier mentioned that HTML tags are not case sensitive. <B> means the same as <b>. The World Wide Web Consortium (W3C), the group responsible for developing web standards, recommends lowercase tags in their HTML 4 recommendation, and XHTML (the next generation HTML) requires lowercase tags.

Tag Attributes

Tags can have attributes. Attributes can provide additional information about the HTML elements on your page. The <tag> tells the browser to do something, while the attribute tells the browser how to do it. For instance, if we add the bgcolor attribute, we can tell the browser that the background color of your page should be blue, like this: <body bgcolor="blue">.

This tag defines an HTML table: <table>. With an added border attribute, you can tell the browser that the table should have no borders: <table border="0">. Attributes always come in name/value pairs like this: name="value". Attributes are always added to the start tag of an HTML element and the value is surrounded by quotes.

Quote Styles, "blue" or 'blue'?

Attribute values should always be enclosed in quotes. Double style quotes are the most common, but single style quotes are also allowed. In some rare situations, like when the attribute value itself contains quotes, it is necessary to use single quotes:

name='Marvin "Wait For It" Eriksen'

Note: Some tags we will discuss are deprecated, meaning the World Wide Web Consortium (W3C) the governing body that sets HTML, XML, CSS, and other technical standards decided those tags and attributes are marked for deletion in future versions of HTML and XHTML. Browsers should continue to support deprecated tags and attributes, but eventually these tags are likely to become obsolete and so future support cannot be guaranteed. For a complete list of tags, visit W3C.org.

Basic HTML Tags

The important tags in HTML are tags that define headings, paragraphs etc.

Tag	**Description**
<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 right aligned heading.</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>

Important: You must indicate paragraphs with <p> elements. A browser ignores any indentations or blank lines in the source text. Without <p> elements, the document becomes one large paragraph. HTML automatically adds an extra blank line before and after a paragraph.

Line Breaks

The <br> tag is used when you want to start a new line, but don't want to start a new paragraph. The <br> tag forces a line break wherever you place it. It is similar to single spacing in a document.
Code:

```
<p>This <br> is a para <br> graph with line breaks</p>
```

Output:
This
is a para
graph with line breaks

The <br> tag has no closing tag.

Horizontal Rule

The <hr> element is used for horizontal rules that act as dividers between sections, like this:

Comments in HTML

The comment tag is used to insert a comment in the HTML source code. A comment can be placed anywhere in the document and the browser will ignore everything inside the brackets. You can use comments to write notes to yourself, or write a helpful message to someone looking at your source code.
Code:

```
<p>
This html comment would
<!-- This is a comment -->
be displayed like this.
</p>
```

Output:
This HTML comment would be displayed like this.

Notice you don't see the text between the tags <!– and –>. If you look at the source code, you would see the comment. To view the source code for this page, in your browser window, select View and then select Source.

Note: You need an exclamation point after the opening bracket <!– but not before the closing bracket –>.

HTML automatically adds an extra blank line before and after some elements, like before and after a paragraph, and before and after a heading. If you want to insert blank lines into your document, use the <br> tag.

Do It Yourself

Open your text editor and type the following text:

```
<html>
        <head>
                <title>My First Webpage</title>
        </head>
        <body>
                <h1 align="center">My First Webpage</h1>
                <p>Welcome to my first web page. 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>
        </body>
</html>
```

Save the page as mypage2.html. Open the file in your Internet browser. To view how the page should look, visit this web page:
`www.cse.iitb.ac.in/~ranveer/tutorials/html/mypage2.html`

Other HTML Tags

As mentioned before, there are logical styles that describe what the text should be and physical styles which actually provide physical formatting. It is recommended to use the logical tags and use style sheets to style the text in those tags.

Detailed information about these tags can be found on W3C.org.

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

Character Entities, Backgrounds, Colors and Lists

HTML Character Entities

Some characters have a special meaning in HTML, like the less than sign (<) that defines the start of an HTML tag. If we want the browser to actually display these characters we must insert character entities in place of the actual characters themselves.

The Most Common Character Entities are less than (&lt; or &#60;), greater than (&gt; or &#62;), ampersand (&amp; or &#38;) etc.

A character entity has three parts: an ampersand (&), an entity name or an entity number, and finally a semicolon (;). The & means we are beginning a special character, the ; means ending a special character and the letters in between are sort of an abbreviation for what it's for. To display a less than sign in an HTML document we must write: &lt; or &#60; The advantage of using a name instead of a number is that a name is easier to remember. The disadvantage is that not all browsers support the newest entity names, while the support for entity numbers is very good in almost all browsers.

Note: Entities are case sensitive.

HTML 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="bg.gif">
<body background="http://cse.iitb.ac.in/~ranveer/tutorials/html/img/bg.gif">
```

The URL can be relative (as in the first line above) or absolute (as in the second line above).
If you want to use a background image, you should keep in mind:

- Will the background image increase the loading time too much?
- Will the background image look good with other images on the page?
- Will the background image look good with the text colors on the page?
- Will the background image look good when it is repeated on the page?
- Will the background image take away the focus from the text?

Note: Later, as you will see, you can use Cascading Style Sheets (CSS) for the purpose of backgrounds.

Colors in HTML

Color Values

Colors are defined using a hexadecimal notation for the combination of red, green, and blue color values (RGB). The lowest value that can be given to one light source is 0 (hex #00). The highest value is 255 (hex #FF).
Visit W3C.org for further details.

Color Names

Only 16 color names are supported by the W3C HTML 4.0 standard (aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow). For all other colors you should use the Color HEX value.

Web Safe Colors

A few years ago, when most computers supported only 256 different colors, a list of 216 Web Safe Colors was suggested as a Web standard. The reason for this was that the Microsoft and Mac operating system used 40 different "reserved" fixed system colors (about 20 each). This 216 cross platform web safe color palette was originally created to ensure that all computers would display all colors correctly when running a 256 color palette.
Again these can be found online.

16 Million Different Colors

The combination of Red, Green and Blue values from 0 to 255 gives a total of more than 16 million different colors to play with (256 x 256 x 256). Most modern monitors are capable of displaying at least 16,384 different colors. To assist you in using color schemes, check out `http://colorschemedesigner.com/`. This site lets you test different color schemes for page backgrounds, text and links.

Lists in HTML

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. An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

Code:

```
<ul>
        <li>Vada Pav</li>
        <li>Pani Puri</li>
</ul>
```

Output:

- Vada Pav
- Pani Puri

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.

Code:

```
<ol>
        <li>Vada Pav</li>
        <li>Pani Puri</li>
</ol>
```

Output:

1. Vada Pav
2. Pani Puri

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

Do It Yourself

Type the following in your text editor:

```
<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 mypage3.html and view it in your browser. To see how your page should look visit this web page:
`www.cse.iitb.ac.in/~ranveer/tutorials/html/mypage3.html`

HTML Links

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

The Anchor Tag and the Href Attribute

An anchor can point to any resource on the Web: an HTML page, an image, a sound file, a movie, etc.
The syntax of creating an anchor:

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

The <a> tag is used to create an anchor to link from, the href attribute is used to tell the address of the document or page we are linking to, and the words between the open and close of the anchor tag will be displayed as a hyperlink.
Code:

```
<a href="http://www.cse.iitb.ac.in/~ranveer">
        Ranveer's Homepage
</a>
```

Output:
Ranveer's Homepage

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.cse.iitb.ac.in/~ranveer target="_blank"'>
        Ranveer's Homepage
</a>
```

Inserting Email Links

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

```
<a href="mailto:ranveer@cse.iitb.ac.in">Email Me!</a>
```

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

```
<a href="mailto:ranveer@cse.iitb.ac.in?subject=Hi!">
        Email Me!
</a>
```

The Anchor Tag and the Name Attribute

The name attribute is used to create a named anchor. When using named anchors we can create links that can jump directly to a specific section on a page, instead of letting the user scroll around to find what he/she is looking for. Unlike an anchor that uses href, a named anchor doesn't change the appearance of the text (unless you set styles for that anchor) or indicate in any way that there is anything special about the text. Below is the syntax of a named anchor:

```
<a name="top">Display Some Text</a>
```

To link directly to the top section, add a # sign and the name of the anchor to the end of a URL, like this:

```
<a href="http://blahblah.com/whatevawhateva.html#top">
        Go back to the top
</a>
```

A hyperlink to the top of the page from within the file 10links.html will look like this:

```
<a href="#top">Go back to the top</a>
```

Note: Always add a trailing slash to subfolder references. If you link like this: href="http://www.cse.iitb.ac.in/~ranveer/tutorials", you will generate two HTTP requests to the server, because the server will add a slash to the address and create a new request like this:
href="http://www.cse.iitb.ac.in/~ranveer/tutorials/"

Named anchors are often used to create "table of contents" at the beginning of a large document. Each chapter within the document is given a named anchor, and links to each of these anchors are put at the top of the document. If a browser cannot find a named anchor that has been specified, it goes to the top of the document. No error occurs.

Images in HTML

The Image Tag and the Src Attribute

The <img> tag is empty, which means that it contains attributes only and it has no closing tag. To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display on your page. The syntax of defining an image:

```
<img src="img/myimg.jpg">
```

Not only does the source attribute specify what image to use, but where the image is located. The above image, img/myimg.jpg, means that the browser will look for the image name chef.gif in a graphics folder in the same folder as the html document itself.

The browser puts the image where the image tag occurs in the document. If you put an image tag between two paragraphs, the browser shows the first paragraph, then the image, and then the second paragraph.

The Alt Attribute

The alt attribute is used to define an alternate text for an image. The value of the alt attribute is author-defined text:

```
<img src="img/myimg.jpg" alt="Kitten!">
```

The alt attribute tells the reader what he or she is missing on a page if the browser can't load images. The browser will then display the alternate text instead of the image. It is a good practice to include the alt attribute for each image on a page, to improve the display and usefulness of your document for people who have text-only browsers or use screen readers.

Image Dimensions

When you have an image, the browser usually figures out how big the image is all by itself. If you put in the image dimensions in pixels however, the browser simply reserves a space for the image, then loads the rest of the page. Once the entire page is loads it can go back and fill in the images. Without dimensions, when it runs into an image, the browser has to pause loading the page, load the image, then continue loading the page. The image would then be:

```
<img src="img/myimg.jpg" width="130" height="101"
 alt="Kitten!">
```

Try inserting this code in the previous DIYs.