

Introduction to HTML



Table of Contents

HTML Introduction	2
What is HTML?	2
HTML Tags	3
Tag Format.....	3
Special Tags	3
Basic Tags.....	4
Comments	4
Useful Tips.....	4
Creating Your First Web Page	6
“Hello World!”.....	6
Breakdown by line	6
Other HTML Elements	8
Character Entities.....	8
Commonly Used Character Entities.....	8
The Anchor Tag	9
Tips on Using Anchors	9
Lists	10
Unordered Lists.....	10
Ordered Lists.....	10
Definition Lists	10
Tips on Using Lists	11
Images	12
Tips on Using Images	12
Tables	13
Tips on Using Tables	14
Creating Valid HTML.....	15
Common Errors.....	15
No DOCTYPE Found!.....	15
No Character Encoding Found!	15
Required Attribute "ALT" Not Specified.	15
Other Resources:.....	16

HTML Introduction

What is HTML?

HyperText Mark-up Language (HTML) is a language designed to create web pages. An HTML file consists of mark-up tags to structure the information on a web page using headings, paragraphs, lists, tables, etc.

HTML was originally defined by Tim Berners-Lee, and then further developed by the Internet Engineering Task Force (IETF). Today HTML specifications are maintained by the World Wide Web Consortium (W3C).

HTML Tags

HTML works in a very simple and logical format. It reads top to bottom, left to right, just like we would read a book in English. HTML is written in plain text with a set of tags to define and distinguish different sections. A basic HTML page can be broken down into a set of headings, paragraphs, lists, tables, etc. Each of these sections is defined by an HTML tag.

Tag Format

An HTML tag is used to define how you want information to be displayed. Each tag begins with a < (less than sign) and ends with a > (greater than sign). What goes between the < and > is the tag. For example, if we wanted to define a paragraph we would use the <p> tag.

Example:

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

Note: at the end there is another tag that closes the paragraph. This is used to identify when the paragraph ends. All tags must have both an open and a close tag. In most cases, the open tag consists of <tag> and the corresponding closing tag consists of </tag>. Some tags you need to define certain attributes and these attributes will go inside the opening tag. We will see this later.

Special Tags

There are some tags that do not need two separate open and a close tags. Instead the tag is defined as <tag />. Notice that the beginning is the same but we simply end it with a /> (back-slash then a greater than sign). This format is used for elements where the tag itself defines the element that is to be used.

Example:

```
<hr />
```

This is the horizontal rule tag. It simply creates a horizontal line that spans across the page. Another commonly used tag of this format is the line break tag.

Example:

```
<br />
```

This is used to end a line when you don't want to start a new paragraph. It forces a line break wherever it is placed.

Basic Tags

Now that you have a general understanding of HTML and how it works, here is a short list of the basic tags you will need to create your first webpage.

Tag	Description
<code><html></code>	Defines an HTML document.
<code><head></code>	Used to enclose other tags which define the characteristics of an HTML page.
<code><title></code>	This is required in all HTML documents. It defines the title of the page.
<code><body></code>	Defines the document's body.
<code><h1></code> to <code><h6></code>	Defines a heading. <code><h1></code> defines the largest heading. <code><h6></code> defines the smallest heading.
<code><p></code>	Defines a paragraph.
<code>
</code>	Defines a single line break.
<code><hr /></code>	Defines a horizontal rule.
<code><!--></code>	Defines a comment.

Comments

A comment is used to insert a note into the HTML source code. All comments are ignored by web browsers. They can be used to explain your code, which can be very useful when you edit your code later on.

Example:

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

Useful Tips

HTML will truncate all the spaces in your code. Any amount of whitespace will count as one. Note: A newline character counts as whitespace.

Using empty `<p>` tags to create blank lines is a bad habit. Use the `
` tag instead.

Introduction to HTML

Jason Smith

HTML automatically adds extra space before and after some elements. For example, space is added before and after paragraphs and headings.

Use lowercase tags. Although tags are not case sensitive, it is recommended to use lowercase tags to prepare yourself for the next generation of HTML. W3C recommends that you use lowercase tags. (XHTML, the next generation of HTML, requires lowercase tags).

Creating Your First Web Page

We now have all the information needed to create our first web page. Being a computer programmer by heart, this is what I call the “Hello World!” of HTML.

“Hello World!”

The “Hello World!” of HTML is simply going to display “Hello World!” on the screen. This will get you familiarized with the fundamentals of creating a web page.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>My First Webpage</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

Breakdown by line

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

This line is the Document Type Definition (DTD), or more commonly referred to as the DOCTYPE. This is required for our code to validate as W3C standard compliant. You should place a DOCTYPE declaration as the very first thing in your HTML document. The DOCTYPE tells a validator which version of HTML to use in checking the document's syntax. We will use this DOCTYPE for our web pages for now and if you want more information on DOCTYPES you can learn about it at:

http://www.w3schools.com/tags/tag_doctype.asp

```
<html>
```

This line defines the start of our HTML document

```
<head>
```

Introduction to HTML

Jason Smith

This line defines the start of the “head information”. A browser does not display this information to the user

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

This line defines the character encoding to be used by a validator. This is required for our code to validate as W3C standard compliant. This line will always go in the head of the document. For more information on character encodings you can learn about it at: <http://www.htmlhelp.com/tools/validator/charset.html>

```
<title>My First Webpage</title>
```

This line defines the title of the document. Each document must have exactly one title defined within the head.

```
</head>
```

This line defines the end of the head of the document.

```
<body>
```

This line defines the start of the body of the document.

```
<p>Hello World!</p>
```

This line defines a paragraph containing “Hello World!”

```
</body>
```

This line defines the end of the body of the document.

```
</html>
```

This line defines the end of our html document.

Other HTML Elements

Now that we have the basics down, let's learn some more things that we can do. Learning HTML from here on is simply learning all the different tags that we can use.

Character Entities

Character entities are special characters that are either non-standard characters, or characters that have special meanings in HTML. For example the < (less than sign) is used to define the start of a tag. If we want to actually display these characters, we have to use a character entity in the source code. All character entities have the same format. They begin with an ampersand (&), followed by the entity name or a # sign followed by a number, and then a semi-colon (;).

Example:

< or < this will display a < (less than sign)

The reason for having two different ways to represent these special characters is that an entity name is easier to remember than a number. A disadvantage to using the name is that not all browsers support the newest entity names. Entity numbers are supported in almost all browsers. A good rule of thumb is to use entity names unless otherwise needed.

Commonly Used Character Entities

Result	Description	Entity Name	Entity Number
"	quotation mark	"	"
&	ampersand	&	&
'	apostrophe	'	'
<	less than	<	<
>	greater than	>	>
	non-breaking space	 	
¢	cent	¢	¢
£	pound	£	£
¥	yen	¥	¥
§	section	§	§
©	copyright	©	©
®	registered trademark	®	®
×	multiplication	×	×
÷	division	÷	÷

The Anchor Tag

The `<a>` (anchor) tag is used to create links to other documents. An anchor can point to any resource that is available on the web: an HTML page, an image, a movie, a Word document, etc. The anchor tag uses the *href* attribute which specifies the address of the destination anchor with a Uniform Resource Identifier (URI).

Example:

```
<a href="http://www.uwaterloo.ca">University of Waterloo</a>
```

To use the anchor tag as a bookmark you must give something on the page the *id* attribute to be able to reference it. For example, if I have a heading at the top of the page I can give it an *id* attribute called *top* to be able to reference it.

Example:

```
<h1 id="top">Top of Page</h1>
...
...
...
<a href="#top">Go to Top</a>
```

Note that you have to precede the name of your bookmark anchor with `#`. Another type of anchor is to use email addresses as the destination. This will open the users default email client with the specified email address as the recipient.

Example:

```
<a href="mailto:someone@somewhere.com">Email Someone</a>
```

For more options that are available for the anchor tag check out:

http://www.w3schools.com/tags/tag_a.asp

Tips on Using Anchors

A link will be displayed in the current browser window unless otherwise specified. To open a link in another window use the *target= "_blank"* attribute.

Do not use "click here" as link text.

If a browser cannot find a named anchor that has been specified, it goes to the top of the document. No error will occur.

Lists

There are three different kinds of lists in HTML: unordered lists, ordered lists and definition lists. Each has their own use depending on what is needed.

Unordered Lists

An unordered list is a list of items marked by bullets. An unordered list is defined by the `` tag and each list item is defined by the `` tag.

Example:

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

Ordered Lists

An ordered list is very similar to an unordered list but is marked with numbers or letters. An ordered list is defined by the `` tag and each list item is defined by the `` tag. The list is automatically numbered by the order in which the list items occur.

Example:

```
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ol>
```

Definition Lists

A definition list is a list of terms and their corresponding definitions. A definition list starts with the `<dl>` tag. Each term starts with the `<dt>` tag followed by one or more definitions for the term defined by the `<dd>` tag.

Example:

```
<dl>
  <dt>Term 1</dt>
  <dd>Definition 1</dd>
  <dt>Term 2</dt>
```

```
<dd>Definition 2</dd>  
</dl>
```

Inside any list item, term or definition you can have other HTML elements such as paragraphs, links, images, etc.

Tips on Using Lists

Use lists for navigation on web pages.

Do not limit yourself to what you can use a list for. When you learn CSS you can style a list to do more than you imagined.

Images

Images or pictures can be used to make web pages stand out more and give the user more visual aids. An image is defined by the `` tag and is another special tag that does not have a separate corresponding closing tag. Instead, it just contains the required attributes needed to define an image and ends with `</>`. The two required attributes for an image are the `src` and `alt` attributes. The image will be placed relative to where the image tag is located in the source code.

Example:

```

```

The `src` attribute defines what image file (.gif, .jpeg, .png, etc) you want to use. The `alt` attribute is used to define “alternative text” for an image that is used when a browser cannot load images. In this case the browser will display the alternative text instead.

Tips on Using Images

Do not go overboard with using images. If you have a page that contains 20 images, the browser has to load each image plus any other files that you may have used.

Try using the same image more than once if possible. Once an image is loaded for the first time it will be kept in cache for a short period of time. It will be able to load much quicker after loading it the first time if it is kept in cache.

Tables

Tables can be very confusing if you do not write them out clearly. They use a lot of tags to create all the rows, columns and cells. Tables are defined using the `<table>` tag. A table is divided into rows and each row is divided into cells. To define a row you must use the `<tr>` tag and to define a cell you use the `<td>` tag. Inside each cell (table data) you can put in any other elements you like.

Example:

```
<table>
  <tr>
    <td></td>
    <th>Monday</th>
    <th>Tuesday</th>
    <th>Wednesday</th>
    <th>Thursday</th>
    <th>Friday</th>
  </tr>
  <tr>
    <th>8:00-12:00</th>
    <td>Work</td>
    <td>Work</td>
    <td>Off</td>
    <td>Off</td>
    <td>Work</td>
  </tr>
  <tr>
    <th>12:00-4:00</th>
    <td>School</td>
    <td>School</td>
    <td>Work</td>
    <td>Work</td>
    <td>Work</td>
  </tr>
  <tr>
    <th>4:00-8:00</th>
    <td>Off</td>
    <td>Off</td>
    <td>School</td>
    <td>School</td>
    <td>Off</td>
  </tr>
</table>
```

As you can see, tables use a lot of code and can be very messy if you do not lay them out clearly in your code. Another tag used in this example is the `<th>` tag. This defines a table heading.

Tips on Using Tables

Only use tables when absolutely needed. Tables use a lot of tags and unnecessary code for small bits of information. Try to use lists or other elements if possible.

Never use tables as a layout manager. It is extremely unnecessary and can become very large. Once you learn CSS it is very easy to make designs without tables.

Creating Valid HTML

An important step in finalizing your web pages is to ensure that the code is valid. This is just a check to see if your syntax is correct and that you haven't missed any of the required tags and attributes. To check to see if your page is valid go to:

<http://validator.w3.org/>

Here you can select your file by its URL address, by uploading the file or by direct input. Either method will give you the same results.

Common Errors

There are many common mistakes you will make from time to time that the W3C validator will catch. Even though when you view your web page it may not look wrong you should still check to see if it is valid. Here are some common errors.

No DOCTYPE Found!

This error occurs when you forget to add the line at the beginning of your file that specifies the document type. To fix this go to your source code and add in the line;

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

You may also want to use a different DOCTYPE and you can learn more about these at: http://www.w3schools.com/tags/tag_doctype.asp

No Character Encoding Found!

This error occurs when you forget to add the line in the head of your file that specifies the character encoding. To fix this go into your code and add the line;

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

You may also want to use a different character encoding and you can learn more about these at: <http://www.htmlhelp.com/tools/validator/charset.html>

Required Attribute "ALT" Not Specified.

This error occurs when you do not specify alternative text for an image. To fix this simply add in the alternative text into the image element.

Other Resources:

<http://www.w3schools.com/>

<http://www.htmlhelp.com/>

<http://www.w3.org/>

<http://www.html-reference.com/>

<http://www.htmlgoodies.com/>

