

Chapter 2: Basics of HTML

Now that we've covered a basic introduction to web programming, let's start learning some actual HTML code. In this chapter, we'll be covering the essentials of HTML. If you are familiar with HTML, you can skip the chapter and go ahead to Chapter 3.

For those of you who are new to HTML, let's get started.

What is HTML

HTML stands for Hypertext Markup Language and is a language used by web programmers to add content to a web page. A markup language is simply a language for annotating a document to explain what different sections of the text are and how they should be presented. For instance, we can use HTML to specify whether the content should be presented as a list or in table form. The current HTML version is HTML5.

The nicest thing about HTML is that the source code of a web page is free for all to view. This makes it easy for us to learn HTML by studying the codes of other web pages. To view the source code of a web page on Windows, simply right-click anywhere on the page and select "View Source" (or something similar, such as "View Page Source", depending on the browser you use). If you are on Mac, click on "View" in the menu bar, select "Developer" and then select "View Source".

Most of the source code that you view will look very complicated, especially if you have no prior knowledge in HTML. Don't worry about that. Soon, you'll be able to write such 'complicated' codes yourself.

To get a better understanding of how HTML5 works, let's start by examining the basic structure of a HTML document.

Basic Structure of a HTML Page

An example of a basic HTML document is shown below. I've added numbers beside each line of the code for easy reference. These numbers are not part of the actual code.

```
1  <!doctype html>
2  <html>
3  <head>
4      <title>My First HTML Page</title>
5  </head>
6  <body>
7      <p>This is just text</p>
8      
9  </body>
10 </html>
```

As you can see from the code above, HTML uses a lot of angle brackets with a single word enclosed inside, such as <head> and <body>. These are known as tags and each tag has a specific meaning in HTML.

Doctype

On line 1, the <!doctype html> tag tells the browser that this document uses HTML5. If you check the source of older web pages, you may see something like <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "<http://www.w3.org/TR/html4/strict.dtd>". This means they are using other versions of HTML, such as HTML4.01 in this case.

Start and End Tags

On line 2, the <html> tag tells the browser that the actual HTML code starts here. Most tags in HTML have a corresponding end tag. The end tag for the <html> tag is found on line 10. It has an additional forward slash (/) before the word html.

Note that not all tags in HTML have end tags. For instance, the tag, which is used to add images to our webpage does not have an end tag. Generally, there is a need for an end tag when we need to let the browser know where the effect of the tag should end. For instance, if we want to bold some text in HTML, we can write

This text is `important`, but this text is not.

We'll get

This text is **important**, but this text is not.

The `` tag and the `` tag tells the browser where the bold effect should start and where it should end. In contrast, there is no need to tell the browser where an inserted image should end. Hence, the `` tag does not require an end tag.

The Head Element

On line 3, we have the start of the head element.

In the broadest sense, a HTML document is made up of two main elements, the head and the body elements.

The head element provides general information about the document, including its meta data, title and links to additional resources. It starts with the `<head>` tag on line 3 and ends with the `</head>` tag on line 5. Within the `<head> . . . </head>` tags, we enclose other tags that provide all these behind-the-scene information about the document.

In our example, we only included information about the title in our head element. The `title` element (on line 4) shows the title that the browser should display on its title bar or on the page's tab. In this case, the text "My First HTML Page" will be displayed. We'll cover more tags that are used within the head element in a later section.

The Body Element

Line 6 is where the body element starts. Contents within the `<body> . . . </body>` tags will be displayed on the webpage. In our example, the text "This is just text" and the image "someimage.jpg" will be displayed.

There are a lot of other tags that we can use inside the `<body> . . . </body>` tags,

such as the `` tag for adding images, the `<table>` tag for displaying tables and the `` tag for adding a list. We'll cover these tags in detail later.

To get a feel of how this works, you can download the code for this chapter from the accompanying website (<http://learncodingfast.com/css>). The source code can be found in the *Chapter 2 - Basics of HTML* folder. Double click on the HTML file to launch it.

You will also be guided through the coding of an actual HTML document when working on the bonus project that comes with this book. The source code for the project can be found in the *Bonus Project\Answers* folder.

Elements Within the Head Element

Now that we understand how HTML works, let us look at the head element in detail.

As mentioned above, the head element provides general information about the document, such as its metadata, title and links to external resources. Let's look at some of the tags within the head element.

`<meta>`

The `<meta>` tag is included within the `<head>...</head>` tags and is used to provide additional information about the website to the browser, search engines or other web services. This information will not be displayed on the page itself. The `<meta>` tag does not have an end tag.

One common use of the `<meta>` tag is to provide keywords for search engines. An example is:

```
<meta name="keywords" content="HTML, CSS, Learn in One Day">
```

You may notice that this tag is a lot longer than the tags we discussed earlier. This is because the `<meta>` tag has two attributes, name and content.

name is used to specify the type of information the tag contains (keywords in this case), while content is used to specify the content of the information.

You can also give a description of your website using the `name=description` attribute. An example is:

```
<meta name="description" content="This is my first Website. It teaches you how to use HTML and CSS">
```

Another common use of the `<meta>` tag is to use it to specify how the browser should control the page zoom level and dimensions. This is done using the `name=viewport` attribute. For instance, you can write

```
<meta name="viewport" content="width=300, initial-scale=1">
```

`width=300` sets the width of the viewport to be equal to 300 pixels. One pixel, `px`, is equal to one dot on the computer screen.

When you set the viewport to 300px and you have an image that is, say, 500 pixels wide, you will only see a portion of the image as the image's width is larger than the width of the viewport. To see the rest of the image, you have to scroll the page. In contrast, if you set the viewport to 500px, the entire image will be shown without any scrolling needed. If you set the viewport to 1000px, the entire image will be shown too, but it'll be smaller and occupy only half the width of the screen.

`initial-scale=1` sets the initial zoom level (1x in this case) of the page when it is first loaded by the browser.

If you are interested in finding out more about how the viewport works, you can check out

<https://developer.apple.com/library/ios/documentation/AppleApplications/Referenc>

`<title>...</title>`

The `<title>` tag is used for defining the title that the browser should display on its title bar or on the page's tab.

`<style>...</style>`

The `<style>` tag is used to add *internal* CSS code to our HTML document. We'll

learn how to do that in Chapter 3.

Example:

```
<style type="text/css">
body {
  ...
}
</style>
```

<link>

The <link> tag is used to link to an *external* resource, most commonly used to link to an external CSS stylesheet. It does not require an end tag.

Example:

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

The `rel` and `type` attributes simply tell the browser that you are linking to a CSS stylesheet. You do not need to modify them. The only attribute that you need to modify is the `href` attribute. This attribute is used to state the path of the CSS file.

How to Write Paths to External Files

The path of any external file always starts from the current folder of the HTML document. Suppose we have five folders: ‘User’, ‘Documents’, ‘MyWebsite’, ‘MyCSS’ and ‘MoreCSS’ with the following structure:

User > Documents > MyWebsite > MyCSS > MoreCSS

That is, the ‘User’ folder contains the ‘Documents’ folder, which in turn contains the ‘MyWebsite’ folder. Within the ‘MyWebsite’ folder, we have the ‘MyCSS’ folder, which contains the ‘MoreCSS’ folder.

If you are working on a HTML file in ‘MyWebsite’ and you want to link to ‘mystyle.css’ in the SAME folder, you simply write `href = "mystyle.css"`.

However, if ‘mystyle.css’ is in the ‘MyCSS’ folder (one level down), you have to write `href = "MyCSS/mystyle.css"`. If it is in ‘MoreCSS’ (two levels down),

you have to write href = "MyCSS/MoreCSS/mystyle.css"

If 'mystyle.css' is in the 'Documents' folder (one level up), you have to use ../ to move one level up. You write href = "../mystyle.css". If it is in the 'User' folder, you have to move two levels up. You write href = "../../mystyle.css".

Elements Within the Body Element

Now that we've covered the elements within the head element, let us move on to the body element.

Elements for Adding Content to the Page

First, let us discuss some commonly used elements for adding content to our webpage. These tags are enclosed within the <body>...</body> tags.

<p>... </p>

This is the paragraph tag and is used to add text to a page. Any content within the two tags will be displayed as a paragraph. By default, most browsers will add a line before and after a paragraph.

Example:

<p>This is a paragraph</p>

This tag is for adding images to your webpage. It requires you to provide some additional information like the location of the image, its height and width *etc.* Commonly used attributes of the tag include:

src:

Stands for "source" and is used to state the path of the image. The src attribute must be provided.

height:

For specifying the desired display height of the image

width:

For specifying the desired display width of the image

alt:

Stands for “alternate” and is used to specify the text to display if the image fails to load.

Example:

```

```

This will insert the image “myImage.jpg” onto the webpage. The image will be scaled to a size of 100px by 100px. If the image fails to load, the text “My Image” will be displayed instead.

<a>...

The <a> tag is used to insert a hyperlink. The most important attribute for the <a> tag is href which is used to specify the URL of the page the link goes to.

Example:

```
<a href="http://www.google.com">Click here to go to Google</a>
```

Output:

[Click here to go to Google](http://www.google.com)

Clicking on the link will bring you to the Google website.

<h1>...</h1> to <h6>...</h6>

The <h1> to <h6> tags are heading tags and are used to define HTML headings. <h1> is the most important heading and <h6> is the least important. Text within heading tags are normally displayed with a larger font size on the browser, with h1 having the largest font size and h6 having the smallest.

Example:

```
<h1>This is the most important heading.</h1>  
<h2>This is the second most important heading.</h2>
```

Output:

This is the most important heading.

This is the second most important heading.

... and ...

The tag stands for ordered list and is used to create a list with numbers or alphabets as list markers.

Example:

```
<ol>
  <li>Chocolate</li>
  <li>Strawberry</li>
  <li>Vanilla</li>
</ol>
```

Output:

1. Chocolate
2. Strawberry
3. Vanilla

... and ...

The tag stands for unordered list and is similar to the tag. However, instead of using numbers or alphabets as list markers, it uses shapes (such as a dot, or a hollow circle).

<table>...</table>, <tr>...</tr>, <th>...</th>, <td>...</td>

The <table> tag is used to create a table. <tr> stands for “Table Row”, <th> stands for “Table Header” and <td> stands for “Table Data”. Tables are created row by row in HTML.

Example (numbers are not part of the code):

```
1 <table>
2   <tr>
3     <th>Name</th>
4     <th>Gender</th>
5   </tr>
```

```
6   <tr>
7     <td>Abigail</td>
8     <td>F</td>
9   </tr>
10  <tr>
11    <td>Benny</td>
12    <td>M</td>
13  </tr>
14 </table>
```

This code will give you a table with 3 rows and 2 columns. Line 2 to 5 defines the first row of the table, which is a header row as the `<th>` tag is used. Line 6 to 9 defines the second row and lines 10 to 13 defines the third. Depending on how you style the table in CSS, you'll get a table similar to the one below:

Name	Gender
Abigail	F
Benny	M

Elements Used in Conjunction with CSS

There are two special HTML elements that do not have any inherent meaning in HTML. They are mainly used in conjunction with CSS to style a specific section of the webpage. These two elements are `div` and `span`.

`<div>...</div>`

`<div>` stands for 'division' and is used to define a division or a section in a HTML document. The `<div>` tag is normally used in conjunction with CSS to format the contents within the `<div>...</div>` tags.

For instance, if we write

```
<div>
  This is some division in the HTML document.
  <ol>
    <li>Chocolate</li>
    <li>Strawberry</li>
    <li>Vanilla</li>
  </ol>
```

</div>

we can use CSS to format everything inside the div tags (i.e. both the text and the ordered list). We'll learn how to do that in the next chapter.

...

The tag is similar to the <div> tag. The main difference is that <div> is a block element, while is an inline element.

A block element is one that starts and ends with a new line break. In contrast, an inline element does not start or end with a line break. For instance, if we write

This is a <div>block element</div>, while this is an inline element.

we'll get

This is a
block element
, while this is an inline element.

As the phrase 'block element' is tagged with the <div> tag, it starts and ends on a new line. On the other hand, the word 'inline' is an inline element and does not start or end on a new line. Generally, we tend to use <div> to wrap sections of a document, while is used to wrap small portions of text, images, etc

Elements For Adding Javascript Code to Website

<script>...</script>

The <script> tag is used to add internal Javascript code to our HTML document or to link to an external script. Javascript is a scripting language that adds interactivity to our website..

Example (to add internal JS code):

```
<script>  
    document.getElementById("para1").innerHTML = "Hello  
JavaScript!";  
</script>
```

Example (to link to an external JS script):

```
<script type="text/javascript" src="myscripts.js"></script>
```

Elements For Formatting Text

...

The tag is used to define important text. Most browsers will bold the text.

Example:

The examination will be held on 12 Jan at 2pm. Latecomers will not be allowed into the hall.

Output:

The examination will be held on **12 Jan at 2pm**. Latecomers will not be allowed into the hall.

...

The tag is used to define emphasized text. Most browsers will display the text in italics form.

Example:

The examination will be held on 12 Jan at 2pm. Latecomers will not be allowed into the hall.

Output:

The examination will be held on 12 Jan at 2pm. Latecomers *will not* be allowed into the hall.

Elements for Defining Sections of a Webpage

HTML also comes with a few tags for defining sections of a webpage. These tags do not do much, their purpose is simply to indicate to the browser and developer which section the content they enclose belongs to.

<header>...</header>

The header element defines the top section of a webpage and normally consists of the logo/banner of the website. Do not confuse the header element with the head element. The head element defines all the behind the scene stuff and is not displayed on the page. The header element on the other hand defines content that is to be displayed at the top of the website.

<nav>...</nav>

nav stands for navigation and is used to define a set of navigation links (i.e. menu).

<main>...</main>

The main element is used to define the main section of a page.

<footer>...</footer>

The footer element is the counterpart of the header element and is used to define the footer of the web page (i.e. the bottom section). The header and footer elements are similar to the 'header' and 'footer' sections of a MS Word document. For Word documents, we normally use the footer to display the page number. On a website, we normally use it to include additional links and additional information (such as contact information and copyright information).

Note that all the four elements above are to be included in the <body> . . . </body>. Their purpose is mainly to further segment the <body> element into different sections. The code below shows how these elements are used.

```
<!doctype html>
<html>
<head><title>An example</title></head>
<body>
  <header>
    <!-- Insert Banner or Logo Here -->
  </header>
  <nav>
    <!-- Insert Navigation Links Here -->
  </nav>
  <main>
    <!-- Insert Main Content Here -->
  </main>
  <footer>
```

```
        <!-- Insert Footer Here -->
    </footer>
</body>
</html>
```

Comments

Notice that in the previous example, we used a lot of the `<!--` and `-->` symbols? These are known as comments.

Most of the time when we program, we need to add comments to our code to make it easier to read and understand. This is extremely important if we are working with other programmers or if we need to edit the source code at a later date. Comments are not displayed in the browser, they are merely added to explain our code.

To add comments to a HTML documents, we use the `<!--... -->` tag.

Example:

```
<!--This is a comment. It will not be displayed in the browser.-->
```

Character Entities

Some characters have predefined meanings in HTML and are reserved for that specific use. For instance, the less than sign (`<`) is used to start all tags. What happens if we need to display the text `5<12` on our webpage?

To do that, we need to use character entities. Character entities always start with an ampersand sign (`&`) and end with a semi-colon (`;`). There are two ways to display the less than sign. We can either write `<` or `<`. The first is known as the entity name while the latter is known as the entity number. An entity name is easier to remember (`lt` stands for less than) but some browsers do not support all entity names. On the other hand, entity numbers are harder to remember but the support is better.

Commonly used character entities include

Less than sign (<)

< or <

Greater than sign (>)

> or >

Ampersand sign (&)

& or &

For instance, if you want to display 5<12 on your website, you write it as 5<12 in your HTML code.

Another commonly used character entity is non-breaking space (). A non breaking space entity is used to display consecutive spaces. By default, HTML does not recognise consecutive spaces. If you write 5 spaces in your HTML code, the browser will remove 4 of them and display only one space. For instance, if you write

“There are 5 spaces here”

the browser will display it as

“There are 5 spaces here”.

If you want to display more than one space, you need to write

“There are 5 spaces here”.