

Coding Made Easy

A Guide & Cheat Sheet

Cara Mooney

CARA MOONEY

CODING MADE EASY

A Guide and HTML Cheat Sheet

Copyright © 2013 by Cara Mooney

All Rights Reserved

Including the right to reproduce, or portions thereof, in any form, under any other name than Cara Mooney. This is solely a resource for reference.

Table of Contents

Foreword	3
Getting Started	3
Font Formatting	4
Bold	4
Italics	4
Underline	5
Strikethrough	5
Subscript and Superscript	5
Comments	6
Blockquote (boxes)	7
Links	8
Images	9
Image Alignment	10
Image Links	12
Paragraphs and Headers	13
Line Breaks	14
Horizontal Rule	14
Lists	15
Bullets (unorganized lists)	15
Numbering (organized lists)	15
Tables	16
Border Attribute	16
Table Headers	17
Meta Tags	18
The Basic Layout of a HTML Webpage	19
Make Your First Webpage	19
Creating Your Home Page	20
The Basic Page Structure	21
Understanding <head> and <body>	21
Doctype	21
All Together Now	23
Quicksheet (review & cheat sheet)	24

Foreword

This is the perfect desktop reference for you when you are updating your website and want a coding cheat sheet of all the basic HTML. This is a good learning tool for young developers to refresh their foundations, but more importantly this e-book was designed for those who are not so savvy to the online world of websites.

There may be times when you need to understand the fundamentals of what makes up your website, even if you are using CMS (content management system) such as WordPress.

WordPress makes things easier in the normal editing of your website, especially when in the Visual mode, but you can always switch into HTML mode and get some practice in! Also, if you see my WordPress e-book, you will learn that sometimes it may be necessary to go into HTML mode so you can work in more detail and add some finesse..

Knowing - or at least recognizing - some of this HTML will really help you, especially if you are *not* using a CMS. This will give you a rundown of all the major and common HTML that you will probably encounter, as well as hints and tips in updating your website. Good luck!

Getting Started

Having a website has so many benefits and can really help you, especially if you operate a business. Of course along with the pluses, come the commitment; the commitment and the code.

HTML is the bones and blood of a website. It is the structural support that allows your website to function. Combined with CSS (cascading style sheets), they create a fully functional and hopefully visually appealing website! Unless you're designing your own entire website, you will probably never have to go into your stylesheet and work with CSS, which basically adds style, color, and flair to the structure that HTML provides.

What is HTML?

Firstly, understand that HTML is the primary mark-up language of the web. It stands for *Hyper Text Mark-up Language*, and is made up of lots of tags and attributes. **Tags** are words surrounded by angle-brackets (< and >) that control the core structure of a page, and attributes 'belong' to tags often allowing further customization or specification of important values. Tags often (but not always) come in pairs, one to open the element and one to close it, but we'll cover this in greater detail in the next tutorial. Think of **attributes** as little add ons.

Font Formatting

This is the most basic and simple of codes. The same options that appear in most emails and text documents are offered for your use on your website. There is Bold, Italics, and Underline in addition to Strikethrough and Subscript and Superscript.

The change from HTML 4 to HTML 5 chanced very little, except they transformed several historically presentational codes and gave them semantic meanings.

Previously, HTML 4.0 used the following for font formatting:

- `BOLD TEXT`
- `<i>ITALIC TEXT</i>`
- `<u>UNDERLINED TEXT</u>`

But now, with the release of the updated, HTML 5, this has changed.

Boldface

```
<strong>The text you want to make bold</strong>
```

Example:

```
<strong>One upon a time</strong> a long time ago there lived a  
fairy princess.
```

One upon a time a long time ago there lived a fairy princess.

Italics

```
<em>The text you want to make italics</em>
```

Example:

```
<em>One upon a time</em> a long time ago there lived a fairy  
princess.
```

One upon a time a long time ago there lived a fairy princess.

Underline

<ins>The text you want to underline</ins>

Example:

<ins>One upon a time</ins> a long time ago there lived a fairy princess.

One upon a time a long time ago there lived a fairy princess.

Strikethrough

<strike>The text you want to strikethrough</strike>

Example:

One upon a time a long time ago there lived a fairy princess.

~~One upon a time~~ a long time ago there lived a fairy princess.

Subscript and Superscript

_{I am so low}

<super>I am so high</super>

Example:

I am normal. I am so low

I am normal. ^{I am so high}

Comments

Comments are hidden in the source code. When a website is “live” or viewed as is, then they are invisible. Only when you go into the code of the website will you see them.

Their uses are diverse. They can be used for a copyright or well...a comment!

```
<!-- COMMENT TEXT -->  
<!-- This is a comment. Comments are not displayed in the browser -->  
<p>This is a paragraph, which will be displayed in the browser.</p>.
```

Often, I will add comments to mark where editing of a HTML document should begin or end. Like so:

```
<!-- START EDITING --> blah blah blah <!-- END EDITING -->
```

Blockquotes

Blockquotes are usually boxes or other design elements (depending on what the website designer established for a boxquote) that allows a portion of text to standout from the rest or the normal text.

```
<blockquote>This text is now a blockquote</blockquote>
```

You could use a blockquote for an announcement or even to display quoted text or a testimonial. Often the designer will have multiple blockquotes, each defined by a class. Note that each class title (in the cases below: announcement, notice, quote) is independent from the code itself. A class title could be “ladeedaaa”, just so long as it has been predefined in the CSS. Ask the designer what predefined blockquote classes have been created for your particular website design.

Example:

```
<blockquote class="announcement">Attention, we have had a sudden  
cancelation due to inclement weather. For further details please  
call our office.</blockquote>
```

```
<blockquote class="notice">Temporarily out of stock.</blockquote>
```

```
<blockquote class="quote">Fantastic! This food is amazing and I  
will definitely be back for more!</blockquote>
```

Take note to use blockquotes sparingly, then there would be little point to use them at all. Blockquotes can be used in place of a paragraph tag and can be placed in between paragraphs. (Ignore the *Loem Ipsum* - faux Latin text that only acts as text filler)

```
<p>Donec ullamcorper nulla non metus auctor fringilla. Fusce  
dapibus, tellus ac cursus commodo, tortor mauris condimentum  
nibh, ut fermentum massa justo sit amet risus.</p>
```

```
<blockquote>Purus Parturient Sem Cursus.</blockquote>
```

```
<p>Etiam porta sem malesuada magna mollis euismod. Nulla vitae  
elit libero, a pharetra augue. Cras mattis consectetur purus sit  
amet fermentum. Nullam id dolor id nibh ultricies vehicula ut id  
elit. Morbi leo risus, porta ac consectetur ac, vestibulum at  
eros. Integer posuere erat a ante venenatis dapibus posuere velit  
aliquet.</p>
```

Links

The HTML `<a>` tag defines a hyperlink. A hyperlink (or link) is a word, group of words, or image that you can click on to jump to another document (webpage/website). The most important attribute of the `<a>` element is the `href` attribute, which indicates the link's destination.

When you move the cursor over a link in a Web page, the arrow will turn into a little hand.

By default, links will appear as follows in all browsers:

- An **unvisited link** is underlined and blue
- A **visited link** is underlined and purple
- An **active link** is underlined and red

This is your link code in its most basic form.

```
<a href="URL">LINKED TEXT</a>
```

Example:

```
<a href="www.google.com">Google</a>
```

- **`target="_blank"`** – You add this into your link so that when clicked it will open into a new window/tap (depending on your browser settings) instead of opening in that window. This is best used when it is an exterior link: when the link is leaving your website to go to an entirely unrelated website. You rarely use this on an interior link: a link that stays within your website.

This is what the link would look like with it included:

```
<a href="URL" target="_blank">LINKED TEXT</a>
```

So you might use the two links like this with the first being an internal link (staying within your website) and the second being an external link (transferring you to an entirely different website).

```
<a href="www.mywebsite.com/about.html">About page</a>
```

```
<a href="www.somerandomsite.net" target="_blank">This Other  
Source</a>
```

Images

Use the `` tag to place an image on your webpage. Like the `
` tag, `` tag does not require a formal ending tag, seen everywhere else. Instead, all we need to do to close this tag out with a slash (/) placed just inside the ending bracket (/>).

This is your image in its most basic form.

```

```

To link an image properly, the ending of the image URL will probably end in an image file name. A few common image files include: **JPEG** or **JPG**, **PNG**, and **GIF**

This is correct:

```

```

This is incorrect:

```

```

Now you can use the image code in its most basic form, but these are some attributes that can make things a little nicer.

- **alt=""** – This stands for *alternative attribute*. Place your text in between the quotation marks. When there is text added, the text will appear when hovered over. Also, if the image has trouble loading or does not load, this would appear in its place. This is required in all images, whether you use it or not in order to validate.
- **title=""** – This is the same thing as an *alternative attribute*, except it is recognized by more browsers. It is good to include this with the same text that you would in the alt. tag. Place your text in between the quotation marks. When there is text added, the text will appear when hovered over. Also, if the image has trouble loading or does not load, this would appear in its place.
- **width=""** – This changes the width of the image (measured in pixels). Place the number in between the quotation marks. If the image is smaller than the number of the width, the image will pixelate and look grainy! Use this only to shrink down larger images. Maintain the same height to width ratio. **Hint:** use width only, and not height, as a way to scale down your images. It will also prevent any distortion of the proportions.
- **height=""** – This changes the height of the image (measured in pixels). Place the number in between the quotation marks. If the image is smaller than the number of the height, the image will pixelate and look grainy! Use this only to shrink down larger images. Maintain the same height to width

ratio. **Hint:** use width only, and not height, as a way to scale down your images. It will also prevent any distortion of the image proportions.

This is what the link would look like with that included:

```

```

So my height is set to 100 pixels, and my width is set to 100 pixels. I have *Alternative Text* in both my alt attribute and title attribute.

Now of course, you don't have to include ALL of that. If you wanted to include just one, this is what it might look like:

```

```

Distortion:

occurs when the proportions of an image are not kept at the same ratio.

To prevent this, find the properly scaled dimensions of the image first or only use one variable (width or height) to define the size.



Pixilation:

occurs when an image that is of a lower quality is scaled up in size.

Never do this with images as it ruins the quality of the image. Only go smaller.



Image Alignment

As far as image alignment, this can be tricky. It would be good to consult with your website designer if this is an option.

In the past there was an *alignment* attribute. It is the simplest solution, and still works with the majority of browsers as well. The newer and more favored solution to alignment is to use a *class*.

```

```

Your options include: *left*, *right*, and *center*

```
align="left"    align="right"    align="center"
```

This will also make it so the text *wraps around* the image, instead of staying in line. Obviously, if the image is centered, no text will be able to wrap around it.



This is where the text is not wrapping around the image....

This is where the text is wrapping around the image. This will occur when the alignment



property of an image is used. For visual purposes, it is always recommended to have text wrap around your images.

The better option is to use a *class*.

```

```

Depending on the designer and how they formatted the CSS, you can put a number of things into the class. Sometimes they will have special formats for certain images even.

Usually I prefer the basic method of alignment class: **class="alignleft"** or right or even center. It will do the exact same thing as the first align attribute, but it this way is much smoother, more modern, and offers more flexibility.

```

```

Class will provide flexibility in the design and visual aspect of the image as well. Ask the designer about what specifics they created. For example, there could be an image class to add a pretty border to any image with that specific class. So instead of saying **class="alignleft"**, you put the other class name.

```

```

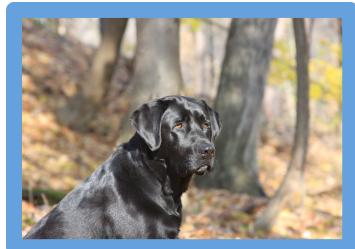


Image Links

Now, you know what a link looks like and you know what an image looks like. Lets combine them! An image can be link, so that when clicked, it goes to a new page/website.

Link code: `LINKED TEXT`

Image code: ``

Image link code:

``

You replace the linked text for the image code, so that you have the link code surrounding the image code.

```
<a href="">~~~~~</a>
    
```

Again, you can add any of the image and link attributes from before. Here they are again.

- **`target="_blank"`**
- **`alt=""`**
- **`title=""`**
- **`width=""`**
- **`height=""`**
- **`class=""`**

Paragraphs and Headers

Text makes up the core content of any web site, and has lots of tags to enable us to display it exactly how we need it.

Paragraph Tags

Paragraph tags structure our paragraphs of text. They tell the browser where a paragraph begins and where it ends which visually separates it from our other content. This stops that “run-on” look that becomes impossible to read. Paragraph tags are one of the easiest to remember, with just the letter “p” inside the < and > symbols. Paragraph tags are closeable elements. Here is an example paragraph:

```
<p>TWO roads diverged in a yellow wood, and sorry I could not travel both and be one traveler, long I stood and looked down one as far as I could To where it bent in the undergrowth.</p>
```

Here is multiple paragraphs:

```
<p>TWO roads diverged in a yellow wood, and sorry I could not travel both and be one traveler, long I stood and looked down one as far as I could To where it bent in the under
```

```
<p> Then took the other, as just as fair, and having perhaps the better claim, because it was grassy and wanted wear; though as for that the passing there had worn them really about the same.</p>
```

```
<p> And both that morning equally lay in leaves no step had trodden black. Oh, I kept the first for another day! Yet knowing how way leads on to way, I doubted if I should ever come back.</p>
```

Headers can break up paragraphs now and again too.

Header Tags

There are 6 levels of header tags, giving us 6 levels of header. This allows us to have a main header, sub-header, sub-sub-header and so on. Although we don’t need all headers, all pages should have at least one... otherwise how will people know what a page or its content is about? The first (and therefore biggest) header is “h1”, and we can create our title inside <h1> tags like so:

```
<h1>My First Header</h1>
```

If we wanted a sub-header further down the page, we’d then use <h2>, <h3> and so forth.

```
<h1>Header 1</h1>
<h2>Header 2</h2>
<h3>Header 3</h3>
<h4>Header 4</h4>
<h5>Header 5</h5>
<h6>Header 6</h6>
```

Now the default settings in any browser have *Header 1* being the largest, with each consecutively shrinking in size. Often, your website design may not follow this premise. Speak with the designer of your website to see which headers are supposed to be entered where.

Other Notes...

Text makes up the core content of any web site, and has lots of tags to enable us to display it exactly how we need it.

Line Break

The
 tag defines a line break in an HTML page. It is as if you went and hit the ENTER key or the RETURN key on your computer. The text would just end and begin on the next line.

In normal HTML, the BR tag would be just
, but with XHTML it needs to be properly closed, and is a self-closing element:

Horizontal Reference <hr />

The <hr> tag defines a thematic break in an HTML page (e.g. a shift of topic) and is an element that is used to separate content (or define a change) in an HTML page, like a line break.

In normal HTML, the HR tag would be just <hr>, but with XHTML it needs to be properly closed, and is a self-closing element: <hr />

Lists

Lists are great for organizing your text. There are two kinds of lists: organized and unorganized. The most common is an unorganized list. They are often coded into your website navigation too!

Unorganized Lists: Bullets

```
<ul>
  <li>Cats</li>      • Cats
  <li>Dogs</li>       • Dogs
  <li>Birds</li>      • Birds
  <li>Fish</li>        • Fish
</ul>
```

The `` tag stands for “unorganized list” and like the majority of the HTML, it opens and ends with the tag ``

The `` tag is the list item. Think of them as bullets. Whatever you wish to bullet must begin with `` and end with ``.

Organized Lists: Numbering

```
<ol>
  <li>Go to the mall</li>    1. Go to the mall
  <li>Pick up the kids from school</li> 2. Pick up the kids from school
  <li>Cash check at bank</li>   3. Cash check at bank
  <li>Make dinner</li>        4. Make dinner
</ol>
```

The `` tag stands for “organized list” and like the majority of the HTML, it opens and ends with the tag ``

The `` tag is the list item. Think of them as bullets. Whatever you wish to bullet must begin with `` and end with ``.

Tables

Tables are very useful and are found often in website code. Unfortunately, tables tend to be on the more intimidating side of basic HTML coding, but we can break this down very easily.

Milk	30%
Flour	22%
Eggs	28%
Vanilla Extract	20%

Tables are defined with the `<table>` tag.

A table is divided into rows (with the `<tr>` tag), and each row is divided into data cells (with the `<td>` tag) `td` stands for "table data", and holds the content of a data cell. A `<td>` tag can contain text, links, images, lists, forms, other tables, etc.

This is your basic table:

```
<table border="1">
<tr>
  <td>row 1, column 1</td>
  <td>row 1, column 2</td>
</tr>
<tr>
  <td>row 2, column 1</td>
  <td>row 2, column 2</td>
</tr>
</table>
```

row 1, column 1	row 1, column 2
row 2, column 1	row 2, column 2

It can get complicated. There are many HTML table generators that you can just input your data and the table code will be created for you (just Google it!).

There are other attributes you can add to the table:

The Border Attribute

If you do not specify a border attribute, the table will be displayed without borders. Sometimes this can be useful, but most of the time, we want the borders to show. To display a table with borders, specify the border attribute. Increase the number, to increase the thickness of the border.

```
<table border="1">
<tr>
  <td>row 1, col 1</td>
  <td>row 1, col 2</td>
</tr>
</table>
```

Other things you can add include along with border are width and cell spacing.

Table Headers

Header information in a table are defined with the `<th>` tag.

All major browsers display the text in the `<th>` element as bold and centered.

```
<table border="1">
<tr>
  <th>Header 1</th>
  <th>Header 2</th>
</tr>
<tr>
  <td>row 1, col 1</td>
  <td>row 1, col 2</td>
</tr>
<tr>
  <td>row 2, col 1</td>
  <td>row 2, col 2</td>
</tr>
</table>
```

Header 1	Header2
row 1, col 1	row 1, col 2
row 2, col 1	row 2, col 2

Meta Tags

Meta tags are important tags that should be included in the <head> of your web page document (that is, between the opening and closing head tags). They generally contain information for the browser – how to render the page, who made it, a description of it, etc. This information is also collected or used by bots when indexing your pages for search engines.

For the most part, the designer will have written out all that you need to have for your meta tags, but it's good to be familiar with them. Some of the information will be reviewed and expressed in more detail to follow. Also more of this is covered in my other e-books.

Meta tags are usually written with “pairs” of attributes — one part telling the browser what the meta tag is about, and the other explaining the content of the meta tag. Types of meta tag are usually defined by the name or http-equiv attributes, depending on whether the meta tag has a name, or is connected with an HTTP header. These are the most common meta tags and how to use them...

Content-Type Meta Tag

The most common — and most frequently discussed — meta tag is content-type. Content-type controls how your pages are delivered and is often used in conjunction with the right DOCTYPE. The most common content-type is text/html delivered with the HTML 4.01 DOCTYPES. Only — because of flaky browser support — XHTML pages with the XHTML DOCTYPE also get delivered as text/html instead of application/xhtml+xml which makes the browser parse the code as HTML making any XHTML pretty redundant (not to mention forgiving: allowing errors that serving a page as XML would not allow). Which content-type you should use to be “right” is a long and often boring road which I really can't be bothered to travel down — I always use text/html irrelevant of the doctype. So, techno-babble aside your content-type meta tag should look a little like:

For HTML users:

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

For XHTML users:

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

(utf-8 is the recommended character set to be used with XHTML)

Description Meta Tag

The meta description tag is quite important in identifying your web page. Admittedly, it doesn't carry the same weight with search engines as it used to, but descriptions ARE searchable. By using a unique and identifying description accurate for each page, you are more likely to rank higher for keywords found in the description/page than if you used a generic description on every page of your website. Search engines often use the listed description underneath a page's title in the results of a search too, so don't forget to make the page sound good!

For HTML users:

```
<meta name="description" content="DESCRIPTION HERE">
```

For XHTML users:

```
<meta name="description" content="DESCRIPTION HERE" />
```

Keywords Meta Tag

Keywords are also dying when it comes to ranking in search engines, but are important in the identification of pages within a website. It hasn't been proven whether all search engines have ditched them in favor of more complex algorithms so add them anyway. You can specify as many keywords as you like (although, apparently, Google blacklists people for "keyword stuffing") Separate each keyword with a comma.

For HTML users:

```
<meta name="keywords" content="KEYWORD1, KEYWORD2, KEYWORD3, ECT.">
```

For XHTML users:

```
<meta name="keywords" content="KEYWORD1, KEYWORD2, KEYWORD3, ECT." />
```

Controlling Bots/Robots

If you're one of those strange people that don't like the idea of search engine robots spidering your web site, or want to block certain pages from being spidered, you need the robots meta tag. Using this meta tag you can control whether or not a page is indexed, and whether or not links on a page are followed.

The default behavior is "*all*" — all pages are indexed and all links followed — but you don't need to specify a robots meta tag just to tell it this. Let's see the possible combinations:

For HTML users:

```
<meta name="robots" content="all,none,index,noindex,follow,nofollow">
```

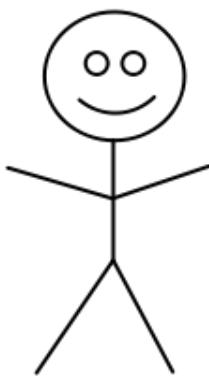
For XHTML users:

```
<meta name="robots" content="all,none,index,noindex,follow,nofollow" />
```

While it is possible to use multiple directives separated by commas as above, it makes no sense to use them all or to have something like "all,none". Unless you want something like "index,nofollow" or "noindex,follow" I suggest ignoring this tag.

The Basic Layout of a HTML Webpage

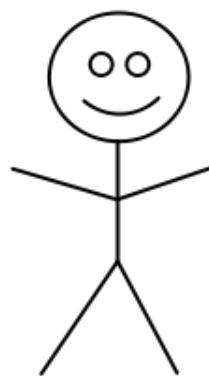
THE BASICS....



```
<html>
<head>
</head>

<body>
</body>
</html>
```

A LITTLE MORE COMPLEX...



```
<html>
<head>
</head>

<body>
</body>
</html>
```

HTML is a great mark-up language because the tags we need to use are often quite obvious in their meaning. For example, the `<html>` tag, which forms the basis of our document, tells the browser that yes — this is an `<html>` document. `<body>` contains the body of the document, ect. You can compare the makeup of an HTML document to that of a person. Each page must have a head and then a body — both integral parts to a human being. We only need one head, and one body, and the head must always come first. Also often included is the *footer*, and even a *container*.

Make Your First Webpage

Every website must start with a home page. This is the main page that your visitors will usually ‘land’ on when they find your website. Although the content of a home page may vary site by site, it is generally where you’d place an introduction to your website and some updates or a blog. This we call our Index page.

Creating Your Home Page

You will need a text editor of some sort so that you can write and read your HTML pages. For a computer using Windows, you will need to open **Notepad** (it is preinstalled). If you are on a Mac, the App store has a number of editor apps that you can download. One I like is **Textwrangler**. *Never* use WordPad or Microsoft Word, or any related program for this.

Open your text editor program to a blank document. Go to *File* and *Save As*. Save this page as **index.htm**. It can also be labeled as **index.html**. There is arguably little difference between the file extensions of htm VS html. Just be consistent. I prefer html though.

The Basic Page Structure

We can apply our basic knowledge of our little HTML person example to create the very simplest page structure.

```
<html>  
  <head>  
  </head>  
  
  <body>  
  </body>  
  
</html>
```

Understanding `<head>` and `<body>`

The `<head>` and `<body>` tags, as well as being important structural tags, have two different functions. Much like a person, the head of an HTML document contains the brains and the body contains the rest.

Inside the head we would put the title of the page, meta tags which control our keywords for search engine robots and when we get more advanced, CSS and JavaScript code. Inside the body we put all of the content: text, images and anything else you may like to see on your web page.

Footers, which are not as important as Headers (`<head></head>`), are where you can usually find the copyright information (i.e. All Rights Reserved © Cara Mooney), maybe a repeat navigation and other miscellaneous information.

Footers may be represented by `<foot>` or `<footer>` or even `<foot-something>`; anything really.

Doctype

A doctype, or *Document Type Declaration*, tells the browser what version of HTML the page is coded in and how to render it. Doctypes go **before** the first `<html>` tag and are the only things that should.

```
<!DOCTYPE blah blah blah>  
<html>  
  <head>  
  
  </head>  
  <body>  
  
  </body>  
</html>
```

Common Doctypes:

Doctypes can get complicated. The simplest option is to stay safe and work with HTML 4 transitional until you know what you are doing.

HTML 4.0 - Transitional

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

HTML 4.0 - Strict

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

XHTML - Transitional

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

XHTML - Strict

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

All Together Now

So basically to put it all together, you have your Doctype at the very top, followed by your opening html tag. Then you have your head, body, and the closing html tag.

```
<!DOCTYPE blah blah blah>  
<html>  
  <head>  
  
  </head>  
  <body>  
  
  </body>  
</html>
```

I know it seems like a lot, especially this last part, but if you break it down, HTML can be very straight forward and easy to comprehend.

This e-book has only shown you the basics. If you wish to learn more and actually create your own webpage, you will need to continue on. Hopefully this has helped you understand HTML a little more or in a little more depth.

Here is an example of that a full page (simplified) may look like, including a footer.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>
    <head> Your meta tags...
        <meta http-equiv="content-type" content="text/html;
        charset=utf-8" />
        <meta name="description" content="" />
        <meta name="keywords" content="" />

        <title>Cara Mooney's Website</title> ←Website title goes here

        <link rel="stylesheet" href="style.css" type="text/css"
        media="screen" /> ←This is to locate your CSS stylesheet
    </head>

<body>

<div id="container"> ←This is common. A “container” to wrap up your content

    <div id="content"> ←This is common as well. Identified your main content.

        <h1>Welcome!!!</h1> ←Your main header

        <p> Praesent commodo cursus magna, vel scelerisque nisl
        consectetur et. Donec id elit non mi porta gravida at eget
        metus.</p>

        <h2>Check Out The Store</h2> ←Another sub-header

        <p> Cras mattis consectetur purus <br />←”Enter”, next line
        sit amet fermentum. Curabitur blandit tempus porttitor.<p>

        <blockquote> Cum sociis natoque penatibus et magnis dis
        parturient montes.</blockquote> ← blockquote

        <p> Sed posuere consectetur est at lobortis. Praesent
        commodo cursus magna, vel scelerisque nisl consectetur.<p>

    </div> ←Ends your content div: <div id="content">

<div id="footer"> ←Starts footer!
    <strong>All Rights Reserved © 2011-2013 <a
    href="http://caramooney.com">Cara Mooney</a></strong>
</div> ←Ends your footer div: <div id="footer">

</div> ←Ends your container div: <div id="container">
</body> ←Ends your body tag
</html> ←Ends your html tag
```

Quicksheet

Bold	<code>TEXT</code>
Italics	<code>TEXT</code>
Underline	<code><ins>TEXT</ins></code>
Strikethrough	<code><strike>TEXT</strike></code>
Comment (<i>invisible code</i>)	<code><!-- COMMENT HERE --></code>
Basic link	<code>LINKED TEXT</code>
Basic image	<code></code>
Basic linked image	<code></code>
Blockquote (box)	<code><blockquote>TEXT</blockquote></code>
Header 1	<code><h1>Header 1</h1></code>
Header 2	<code><h2>Header 1</h2></code>
Header 3, and so forth...	<code><h3>Header 3</h3>, <h4></h4>, ect...</code>
Paragraphs	<code><p>Paragraph text....</p></code>
Line Break (“Enter”)	<code>
</code>
Horizontal Rule (“Bar”)	<code><hr /></code>
Unordered List <i>(with bullets)</i>	<code> Bulleted item 1 Bulleted item 2 Bulleted item 3 Bulleted item 4 </code>
Ordered List <i>(with numbers instead of bullets)</i>	<code> Bulleted item 1 Bulleted item 2 Bulleted item 3 Bulleted item 4 </code>