Feb 20, 2015
CheckoutCrypto Canada

Introduction to HTML + CSS

By: Grant Hutchinson

Feb 20, 2015
CheckoutCrypto Canada

Index

Feb 20, 2015
CheckoutCrypto Canada

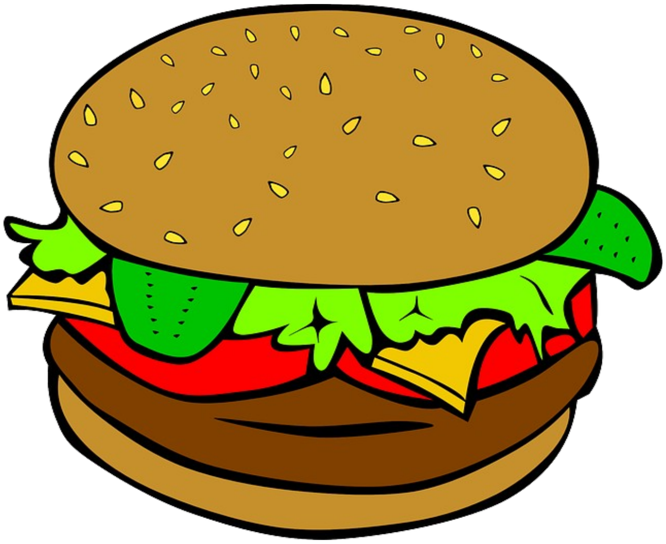
HTML

HTML stands for HyperText Markup Language, it's a web language specifically for authoring pages. Each seperate tag of code is known as an element. They begin and end with brackets e.g. <p> and ends with brackets and a backslash </p>.   While some tags are self closing e.g. <img />

Think of each section of elements as building(lego, brick) blocks, or like burgers(<body></body>), stacked ontop of and within each other, all inside a big bag(<html></html>). Each line within an element is indented, with each proceeding line following indenting further.

Any browser can read html pages as well as a large number of scripts, images, resources they access. HTML is standard markup for anything that needs to be displayed on a page. Even if you choose to use another language for your web page, you'll still end up displaying and utilizing HTML.

Basic HTML structure:



```
<html> <!-- The Hamburger Bun Top-->

        <head><title>Learn To Code Example</title></head> <!-- Lettuce, Cheese, Tomato-->

        <body>          <!-- Burger Patty-->

                <p> Hello World! </p>   <!-- Ground beef, spices, sauce-->

        </body>

</html><!-- The Hamburger Bun Bottom-->
```

Explanation:

Every element you see is necessary to complete a basic html page(except for the <p>Hello World!</p>)

The <html> and </html> elements open and close the start and end of the html page.

<head> and </head> tags open and close the page header element, where scripts are other things are placed before the page(body) loads. The <title> element is for the page title you see ontop of your browser window

<body and </body> tags open and close the body element of the page. All page content goes between this element.

<p> and </p> open and close a paragraph.

<!-- and --> are commented text. They will be visible in page source, not on page display, in browser.

Try it!

Open a text editor, **type** out the example structure above. Make sure you're careful about every single character. Spacing doesn't matter, for your own testing. However, if you want anyone to read your code, I recommend following a similar indenting style. Always space at least one tab in when working within a structure of code or stanza. It makes it easier for you and everyone else who will read your code.

Save the file to your site's root /var/www/site (whatever you called it), as practise.html

Open a browser, visit: http://127.0.0.1/site/practise.html (whatever your site root folder is instead of “site”). You should only the text:

Hello World!

note: You may also choose to do this practise from a file, placed anywhere, instead of using an apache server andvisiting the page. Instead save the file anywhere, right click the saved file and open with browser.

Congratulations, you have written your first html web page. Now you're ready to begin learning the wonderful world of web programming. I'm not going to sugarcoat it, their is a lot of practise, hard work, time, involved with gaining a good grip on web programming. Many make their living studying every aspect about it. This is your introduction to that world. I can only lead you to the water, you have to drink for yourself.

Let's learn some basic CSS at the same time. I think it's important to understand, while these two markups are different, they both function together. Thus you might aswell learn them both at the same time, you will be using both in any given web programming project.

CSS

CSS stands for Cascading Style Sheets, it's the designer's template, for how they want the HTML elements to layout. The main reason for CSS is simple and clear, immediately when you begin, to keep the style, presentation, layout, scale, within a templated design .css file. This way, we can easily edit layout changes without having to change the HTML blocks. CSS gives you the ability to effect these blocks in infinite ways, rather than be stuck with one static style. This is especially useful for building esthetic, dynamic, scalable, pages. It's a standard, every programmer must know CSS.

Chrome and other browsers allow you to view the CSS markup for any given page, modify elements live within the browser, making it much easier to debug and place elements within a layout. Recommend you test right clicking a page in chrome click “inspect element”. Then you can select the HTML element from the left panel(under elements tab), then edit the CSS in the right panel(styles tab).

Basic CSS Structure:

```
body {
	background-color:green;
	color:white;
}

p {
	color:black;
}
```

Explanation:

body is the main html structure element tag, one of the only included in our basic example structure. By including a body element, we're telling css to apply these design rules to anything within the <body></body> element. That is, anything that doesn't already have a specific design rule designated for it.

p is the paragraph element tag, one of the only included in our basic example structure.
By including a p element, we're telling css to apply these design rules to any paragraph(<p> </p>) on our page. That is, anything doesn't already have a specific design rule designated for it.

This is just an example with two basic, standard elements. In CSS, you can cite IDs and classes from any element or series of elements or structures, including different screen configurations.

Try it!

Open a text editor **type** out the CSS in the structured example above. Save it it to your site's root folder /var/www/site/style.css. Open the practise.html file you wrote earlier, add the following between the <head></head> structured element. <link href="./style.css" rel="stylesheet" />

Lesson One: text elements, ordered and unordered lists

Copy your practise.html and rename it: practise_1.html.

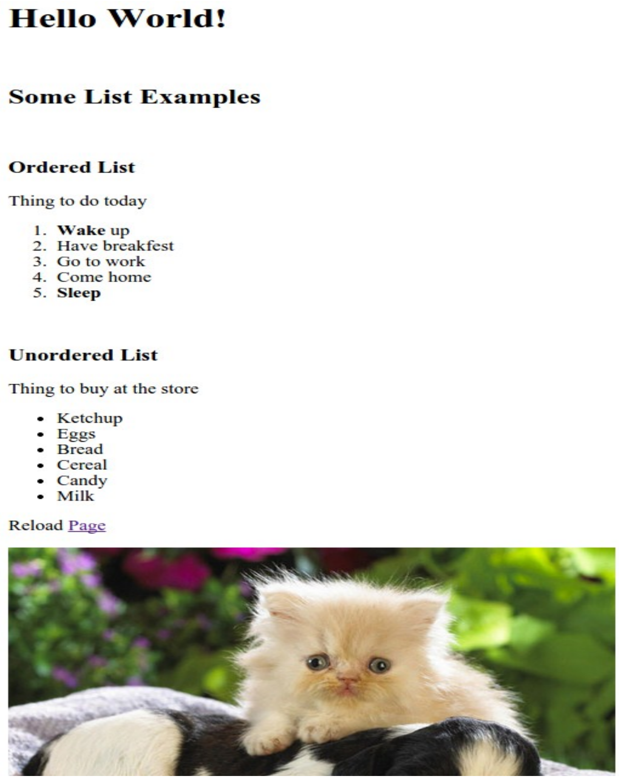
Edit the html(hamburger) structure, by adding to the contents of your body(beef) with the following, recommend you **type** it out. Typing things out always helps for memory and you will gain little benefit unless you learn it correctly. Place between <body></body> tags of practise_1.html

```
<h1> Hello World! </h1>
<br>
<h2>Some List Examples</h2>
<br>
<h3> Ordered List </h3>
<p>Things to do today</p>
<ol>
        <li> <b>Wake</b> up </li>
        <li> Have breakfest </li>
        <li> Go to work</li>
        <li> Come home </li>
        <li> <b>Sleep</b> </li>
</ol>
<br>
<h3> Unordered List </h3>
<p>Things to buy at the store</p>
<ul>
        <li> Ketchup </li>
        <li> Eggs </li>
        <li> Bread </li>
        <li> Cereal </li>
        <li> Candy </li>
        <li>Milk</li>
</ul>
<p>Reload <a href="./html_test.html" > Page </a></p>
<img src="http://i.imgur.com/reS2PeL.jpg" width="500px" height="420px">
```

Explanation:

<p></p> - Paragraph element
<b></b> - bold element
<h1></h1> <h2></h2>,<h3></h3>, - heading, with each number higher than 1 being a smaller sub-heading
<ol></ol> - ordered list
<ul></ul> - ordered unordered lists
<li></li> - list items
<a href> </a> - a link, usually contains a source as well as a target anchor,
<img /> - an image link. Usually, it's best to define either in the CSS or html, the width and height for the image, in addition to the image URL.
<br> **-** page break

<u>Try it!</u>
Open a text editor, open the practise_1.html file, **type** out the HTML in the structured example above. Save it it to your site's root folder /var/www/site/practise_1.html Right click the file, open with browser.



Hello World!

Some List Examples

Ordered List

Thing to do today

1. **Wake** up
2. Have breakfest
3. Go to work
4. Come home
5. **Sleep**

Unordered List

Thing to buy at the store

- Ketchup
- Eggs
- Bread
- Cereal
- Candy
- Milk

Reload <u>Page</u>

<u>Lesson Two: tables and divs</u>

Copy your practise.html and rename it: practise_2.html.

Edit the html(hamburger) structure, by adding to the contents of your body(beef) with the following, recommend you **type** it out.

```
<html>
 <!-- The Hamburger Bun Top-->
        <head>
                <title>Learn To Code Example</title>
                <link href="./style.css" rel="stylesheet" />
        </head> <!-- Lettuce, Cheese, Tomato-->

        <body>          <!-- Burger Patty-->
                <table width="100%" >
                        <tr> <td colspan="2"><h3>Eating a hamburger</h3></td>
                        </tr>
                        <tr>
                                <td> <b><u>Positives</u></b> </td>
                                <td> <b><u>Negatives</u></b> </td>
                        </tr>
                        <tr>
                                <td>Filling</td>
                                <td>Tons of Calories </td>
                        </tr>
                        <tr>
                                <td>Relatively inexpensive </td>
                                <td>increases garbage in environment</td>
                        </tr>
                        <tr>
                                <td>Quick </td>
                                <td>bad for your health</td>
                        </tr>
                </table>
        </body>

</html><!-- The Hamburger Bun Bottom-->
```

Explanation:

<table></table> -table element tag, each table structure must begin and end with this tag. Parameter you may wish to add include, height, width, border, background, etc.
<tr></tr> - new table row. If you want a row to stretch we use rowspan.
<td></td> - new table column. If you want a col to stretch we use colspan

Try it!

Open a text editor, open the practise_2.html file, **type** out the HTML in the structured example above. Save it it to your site's root folder /var/www/site/practise_2.html Right click the file, open with browser.

Eating a hamburger

Positives	**Negatives**
Filling	Tons of Calories
Relatively inexpensive	increases garbage in environment
Quick	bad for your health

Part Two:

Copy your practise.html and rename it: practise_3.html. **Also** copy style.css and rename it style3.css

Edit the html(hamburger) structure, by adding to the contents of your body(beef) with the following, recommend you **type** it out.

```
<html>
 <!-- The Hamburger Bun Top-->
        <head><title>Learn To Code Example</title>
                <link href="./style3.css" rel="stylesheet" />
        </head> <!-- Lettuce, Cheese, Tomato-->

        <body>          <!-- Burger Patty-->
                <div class="main">  <!-- Total Outter page box -->
                        <div class="header"><p>Header/logo here</p></div>
                        <div class="navigation"><p>Navigation bar here</p></div>
                        <div class="content"><p> Some content here</p></div>
                        <div class="footer"><p>Footer here</p></div>
                </div>
        </body>

</html><!-- The Hamburger Bun Bottom-->
```

Explanation:

<div></div> - the layout structure block that defines a section or space. We identify these blocks with names called classes. We use these classes in the CSS to style our page.

Change your style3.css to the following:

```
/*
*  Remember, each class inherits the parameters of the class it resides in.
*/


.main {
            width:1024px;
}

.header {
            height:12px;     /// used to be a rule of thumb, do not make the header/navigation too large
            color:teal;
}

.navigation {
            height:12px;     /// used to be a rule of thumb, do not make the header/navigation too large
            color:red;
}

.content {
            height:300px;
            color:white;
}

.footer {
            height:12px;     /// used to be a rule of thumb, do not make the header/navigation too large
            color:grey;

}
```

Try it!

Open a text editor, open the practise_3.html file, **type** out the HTML in the structured example above. Save it it to your site's root folder /var/www/site/practise_3.html Right click the file, open with browser.

Header/logo here

Navigation bar here

Some content here

Footer here

Notice anything? Our blocks are stacked, not side by side why is that? Why does each block extend to the width of the page? Well that's the default 100% of the page for any given div. If we want them to be beside each other, with different sizes, etc, we have to utilize our CSS file to style, float, scale the blocks to our choosing.

“But I want to make a cool layout not a bunch of stacked **ugly** blocks!”

No problem, our next part of the tutorial goes into how to style a basic web page template, for use with our final tutorial for the day in PHP. You needed to understand how divs layout by default first, before you can modify them and build a proper design.

Lesson Three: HTML Forms

Copy your practise.html and rename it: practise_3.html Open it and replace the contents of the <body></body> with the following:

```
<div class="content">
        <form action="practise_3html" method="get">
                <p> Enter your name: </p>
                <input type="text" name="name" />
                <input type="hidden" name="action" value="name" />
                <input type="submit" text="submit" />
        </form>
</div>
```

Explanation:

<form> </form> - This form, within a div content block, will display a text field for input, along with a submit button. The action section is the URL to which the form will be sent, the method is either get/post.

<input /> - this is the input field element, their are several types, including: text, checkbox, radio, hidden, submit. You can also assign element id and classes to inputs and forms.

Note: this tutorial does not deal with php and javascript. I'm not going to get into it, basically from this form, you would continue, by having a clientside(javascript) and server side(PHP) validation. Neither of which are covered in this tutorial.

Try it!
Open a text editor, open the practise_3.html file, **type** out the HTML in the structured example above. Save it it to your site's root folder /var/www/site/practise_3.html Right click the file, open with browser.

Enter your name:

Lesson Four: HTML + CSS Building a basic page template

Take the entire lesson's tutorials, combined, this is what you get. When designing a website, ideally you should start by building the basic layout(as we did above in lesson two) from which to fill each page. Section it into one overall page class, so that everything within that page, will scale to the page size. Thus making it easier when you're designing for mobile, desktop, scale environments. Within that one overall page class, subsection it into: header, navigation, content, footer, etc.

To begin a new html file called template.html and a style called style_template.css. Place the following in the template.html file and save.

```
<html><!-- The Hamburger Bun Top-->
          <head><title>Learn To Code Example</title>
              <link rel="stylesheet" type="text/css" href="style_template.css" />
           </head>
<body>   <!-- Burger Patty-->
          <!-- Begin Head_menu -->
   <div class="page">
       <div class="header">
       <div class="logo"><img src="http://i.imgur.com/O5nB7Q8.jpg" width="300px"></div>
        <div class="head_menu">
             <table class="head_menu_layout">
                <tr>
                   <td class="head_menu_itm"><a href="somepage1">item1</a></td>
                   <td class="head_menu_itm"><a href="somepage2">item2</a></td>
                   <td class="head_menu_itm"><a href="somepage3">item3</a></td>
                   <td class="head_menu_itm"><a href="somepage4">item4</a></td>
                   <td class="head_menu_itm"><a href="somepage5">item5</a></td>
                </tr>
             </table>
     </div>
<!-- End Head_menu -->
</div>

<!-- Begin Content -->
<div class="content">
```

```
<p> We can put anything we want in here.  This page is preset for 720px height minimum, it will scale to larger sizes.</p>

</div>
<!-- End Content -->
</div>
<!--- Begin Footer -->
   <div class="footer">
      <table class="foot_layout">
         <tr>
            <td class="foot_layout_item">CheckoutCrypto (2015). </td>
            <td class="foot_layout_item">CheckoutCrypto Developer Sample. </td>
            <td class="foot_layout_item"><a href="http://www.gnu.org/copyleft/gpl.html" >GPLv3 Public License</a></td>
         </tr>
      </table>
   </div>
   </div> <!-- End My Page Template -->
</body>
</html><!-- The Hamburger Bun Bottom-->
```

Place the following in a new file called style_template.css file :

```
/*
*Copyright 2015 CheckoutCrypto and CheckoutCrypto Canada GPLv3 Licensed template
*/

.page {
   height: 720px;
}
/* MAIN HEADER */
.header {
   height:60px;
   background-color:white;
         border-bottom:5px solid #1E91F3;
}
.logo {
   float:left;
   margin-left:50px;
   margin-top:5px;
   height:50px;
   width:120px;
```

```
}
.head_menu {
    padding-left:10px;
    margin-left:200px;
    float:left;
    margin-top:20px;
    height:30px;
    width: 55%;
    color: green;
}
.head_menu_layout {
   width:100%;
}
#a.head_menu {
    color:red;
}
.head_menu_itm {
     width:200px;
}

/* MAIN CONTENT */
.main {
}
.menu {
     width:150px;
     background-color:red;
     height:300px;
}
.content {
    background-color:white;
    min-height:150px;
    margin-left:25px;
}

/* MAIN FOOTER */
.footer {
    height:40px;
    color: blue;
    background-color:white;
}

.foot_layout {
     border-top:1px solid #E5E5E5;
     padding-top: 10px ;
    padding-left:60px;
    height:100%;
    width:100%;
}

.foot_layout_item {
     width:35%;
}
```

Try It!

This is the same template used for CheckoutCrypto's Developer tools. You're welcome to modify and learn from it. This concludes the tutorial, suggest you do further reading into the specifics of CSS or HTML attributes, in order to fully understand what's going on in the design above.