

User Interface and Web Design - Take-home Booklet

Take Home Item: A website all about something you care about!

What you will learn: User Interface, HTML and CSS.

June 24, 2017

HP Campus in Corvallis, OR

Exercise 1: Let's Begin

How is a web page structured?

Look at your index.html code that you got from the flashdrive.

- The DOCTYPE declaration defines the document type
- The text between `<html>` and `</html>` describes the web page
- The text between `<body>` and `</body>` is the visible page content
- The text between `<h1>` and `</h1>` is displayed as a heading
- The text between `<p>` and `</p>` is displayed as a paragraph
- What's all that indentation? The indentation helps you keep open and closed tags organized. It's easy to miss a closed tag if your indentations aren't correct and when that happens, your web page may not render as you expect.
- HTML is the base of the website, and you can link to stylesheets (CSS files) and script files (Javascript) so the elements on in HTML file display with styles (from your CSS files) and programmed functionalities (from your Javascript files)

What is HTML?

- HTML is a language for describing web pages
- HTML stands for Hyper Text Markup Language
- HTML is a markup language
- A markup language is a set of markup tags
- The tags describe document content
- HTML documents contain HTML tags and plain text
- HTML documents create web pages

HTML Tags

- HTML markup tags are called HTML tags
- HTML tags are keywords (tag names) surrounded by angle brackets like `<html>`
- HTML tags normally come in pairs like `<h1>` and `</h1>`
- The first tag in a pair is the "opening tag", the second tag is the "closing tag"
- The closing tag is written like the opening tag, except with a forward slash before the tag name

Exercise 2: Let's Add More Stuff

HTML Hyperlinks (Links)

The HTML tag defines a hyperlink. Adding a hyperlink (or link) to an element on a page makes it clickable. When you move the cursor over a link in a Web page, the arrow will turn into a little hand. The most important attribute of the element is the href attribute, which indicates the link's destination. By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple

You can test this out by Googling something and looking at what colors the links you've visited are and what color the ones you have are.

HTML Link Syntax

The HTML code for a link is simple. It looks like this:

```
<a href="http://www.google.com">Link text</a>
```

The href attribute specifies what URL you want the link to go to. The <a> tag

Example

Let's add a link to our existing website.

1. Open your HTML file in Cloud9.
2. Add the following anywhere you'd like between the <body> </body> tags:

```
<a href=" http://www.google.com">  
    Where should we go?  
</a>
```

3. Save the file and refresh your HTML page in the browser
4. Click the new link on your webpage and see what happens. Cool!
5. Now, let's modify that link so the link opens in a new tab with someone clicks on it

Change

```
<a href=" http://www.google.com">
    Where should we go?
</a>
```

To

```
<a href=" http://www.google.com" target="_blank">
    Where should we go?
</a>
```

6. Save the file and refresh your HTML page in the browser

7. Click the new link on your webpage and see that the link opens in a new tab

What if we changed the link from text to an image? Let's try it!

1. Do a Google Image search for an image you like.

2. Right-click and select "Save Image As...", navigate to your directory, and save image as myImage.png

3. Return to your HTML file and change

```
<a href=" http://www.google.com" target="_blank">
    Where should we go?
</a>
```

To

```
<a href="http://www.google.com" target="_blank">
    <img src=myImage.jpg />
</a>
```

4. Open your HTML file and right below this code

5. Save the file and refresh your HTML page in the browser. So you see your image?

Now it's time to add some colors and borders to your webpage using CSS.

Exercise 3: Time for Some Color

What is CSS?

-- CSS stands for Cascading Style Sheets, and is how you add style to elements on your page (remember that elements are indicated using <> tags)
-- You put your CSS rules inside curly braces {}, with each rule followed by a semicolon, like this:

```
h1 { color: red; }
```

Note: Did you notice how the <h1> text already is bigger than the <p> text? That's because browsers automatically make text inside the <h1> tag bigger than text inside the <p> tag (hence why it's called a "heading"). So no need for CSS there!

1. Open a new text file in the same folder as your HTML file and name it styles.css
2. Type the following

```
h1 { color: red; }
```

<http://www.color-hex.com/> is a good source for colors and their hexadecimal notations. You can use the hexadecimal code or, for common colors, just use their name.

Let's tell our HTML about the CSS!

3. In your HTML file, in between the <head> tags, on a new line add the following to let your HTML file know it should apply the styles in your CSS file:

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

4. Your HTML should look like the following, plus whatever paragraphs or headers or other elements you've added:

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="styles.css" >
  </head>
  <body>
    <a href="http://www.google.com" target="_blank">
      <img src = myImage.jpg></img>
    </a>
```

```
<h1>This is a header</h1>
<p>This is a sentence .</p>
</body>
</html>
```

5. Save both files and refresh your web page to see your changes.

Exercise 4: Let's Continue to Enhance Our Webpage

1. In your CSS, add the following styles to the `<p>` tag. These will apply to every `<p>` tag in your HTML.

```
p {
    margin-left:50px;
    color:blue;
}
```

This will add 50 pixels of margin on the left side of your text, and turn the text blue.

2. Save both files and refresh your web page to see your changes. See how the text moved over to the right? That's because `margin-left` adds space to the left of whatever element it's applied to, and in this case the margin is on the left side, so the text gets pushed to the right.

If you want to add more styles to your page, check out this list of basic CSS properties to get started: <http://www.zell-weekeat.com/9-important-css-properties-you-must-know/>. In addition, you can add more selectors (for example, lists or different sized headers). Here are some to get started with: http://www.w3.org/wiki/CSS_basics#Defining_style_rules.

It's time we add some dynamic content to our webpage, using the programming language Javascript!

Exercise 5: JavaScript

1. In your HTML file, inside the `<body>` tags, on new line add the following:

```
<h1>My First JavaScript</h1>
<p>Click the button to display the date.</p>
<button type="button" onclick="dateFunction()">What is the
date?</button>
<p id="demo"></p>
```

3. There's some new stuff here! The `<h1>` header tag and the `<p>` paragraph tag we are familiar with. But, what's that "id" in the paragraph tag and now we have a button? In Javascript, there is a `getElementById()` method that we can use to do things to only that element. And the button tag is just that, a button! When the button is clicked, the `onclick()` event is fired (because the button has been clicked) and the `dateFunction()` function is fired.

4. Create a new document in the same folder as your HTML and CSS files, and this time call it `scripts.js`. On the first line of the file, type:

```
function dateFunction()
{
    document.getElementById("demo").innerHTML = Date();
}
```

Congratulations! You just wrote Javascript!

5. In your HTML file, right above the closing `</body>` tag, add the following to tell the page to apply the Javascript in the `scripts.js` file:

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

6. Save your Javascript and HTML file and refresh the page to see your changes. Now you can click the "What is the date?" button and the current date will appear.

How does this work? When you click the button, the `dateFunction()` is fired in the Javascript file. In this example, the function finds the HTML element with an `id` equal to `demo` and sets the text of it to be the current date. `Date()` is also a function, it just happens to be a function built into JavaScript and the browser knows how to get the current date, which is why you don't have to tell the browser what to do with it.

Exercise 6: Alert The World That You Learned JavaScript

1. In your Javascript file, right below the closing bracket of your `dateFunction()`, add the following:

```
function helloWorld()  
{  
    alert("I just wrote Javascript!");  
}
```

3. In your HTML, under the button tag that displays the date when clicked, type this:

```
<button onclick="helloWorld()">Click me!</button>
```

4. Save your Javascript and HTML file and refresh the page to see your changes.

5. Click the “Say Hello to The World!” button to see what happens. How does this work? Well, as with the Date button, when you click the “Say Hello to The World!” button, the `onclick()` event is fired (because the button has been clicked) and the `helloWorld()` function is fired. The `alert()` function is built-in like the `Date()` function and when it’s fired, it fires an alert.