

Lab 1: Introducing HTML5 and CSS3

In this lab we will cover some basic HTML5 and CSS, as well as ways to make your web app look and feel like a native app.

Your task:

Create an HTML5 web app for Bike Drexel that looks and feels like a **native** iOS app.

To start with, your app should consist of 2 files: **index.html** and **iphone.css**. The content of index.html can be copied from the end of this document.

Your app should open from the Home screen of your device.

Step-by-Step Tutorial

Step 1: Create a project directory for this lab. Open Sublime Text (or your favorite text editor) and create two files: `index.html` and `iphone.css`.

Step 2: Grab the images from Drexel Learn in the Lab1 assignment and save them in your project directory.

Step 3: Fill in the content of index.html by copying and pasting the [HTML 5 code at the end of this worksheet](#).

Step 4: Study the HTML code.

When you're done, save your work. To view your work so far, click on the index.html in your project directory. A web browser will open and things should look something like this:

Bike Drexel



Welcome to Bike Drexel

We believe in bicycles! Everything you need to be a Drexel Cyclist.

- [Bike Stores](#)
- [Bike Maps](#)
- [Go for a Ride](#)

CS530: Developing User Interfaces © 2017

Step 5: Next, let's add some style rules for your app using CSS.

If you are new to CSS, check this [tutorial on CSS syntax](#).

The first thing to do is attach a stylesheet to your **index.html** file. You can do that by adding this line in the `<head>` section of the file.

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

Also, this next line makes your web app look a bit more native on an iPhone, but you'll need a device to test it. Make the following additions and changes to the next line, so that it looks like this:

```
<meta name="viewport" content="width=device-width, user-scalable=no" />
```

Step 6: Now, working in **iphone.css**, let's define some fonts and colors for your app.

Start with the **body** and **header** style:

```
header {  
  font-family: Futura; Trebuchet MS;  
  font-weight: bold condensed;  
  color: #07294D;  
  font-size: 25px;  
}
```

Define your own rules for styling `<body>` and `<h1>`.

For information about CSS colors check [this tutorial](#).

If you are interested, here are the colors that Drexel uses on official websites:

<http://drexel.edu/identity/web/colors/>

Checkpoint: To test your styled app, you will need to use a device or a simulator:

To test using your device:

- 1) Use Fetch or another program to access your CS tux server account. If you are not on the Drexel network, you may need to use Cisco VPN to connect to the network first. Also, if you have problems connecting, make sure that you are using SFTP.
- 2) Inside your public_html directory create a new directory; name it cs338.
- 3) Transfer your files into your cs338 directory.
- 4) Now, using your device, access your web app through the browser by typing the URL for your cs338 directory.

Now your page should look something like this:



Step 7: Continue to style your app by defining some [padding](#). Add the following code to the `iphone.css` file.

```
#content {  
    padding: 3px;  
}
```

Add padding rules to `<p>` `<h1>` and `<header>`

Also, we can center our image:

```
#Drexel{  
    /*center the image*/  
    display: block;  
    width: 100%;  
    margin-left: auto;  
    margin-right: auto;}
```

Step 8: Now we can add rules to make the navigation items look more like native iPhone apps:

```
#main_nav ul { padding: 0; font-weight: bold; }  
  
#main_nav li a{  
    display: block;  
    margin: 6px;  
    padding: 9px;  
    border-radius: 10px; /* rounded corners */  
    list-style-type: none;  
    text-decoration: none;  
    color: #222;  
  
    /* nice gradient shade on the buttons */  
    background-image: -webkit-linear-gradient(top, #f8f8f8 0, #d4d4d4 100%);  
    background-image: linear-gradient(to bottom, #f8f8f8 0, #d4d4d4 100%);  
  
    border: 1px solid #666; /* dark outline */  
}
```

Step 9: Save all of your work and refresh your web page to view your work. Now, let's also add some rules to change the default behavior of the buttons in our app so that you get visual feedback on push events.

```
nav li:active {  
    background-color: #ccc;  
    background-image: none;  
    box-shadow: inset 0 3px 3px rgba(0, 0, 0, .2);  
}
```

Step 10: Finally, style the `<footer>` tag .

Checkpoint: Test your web app on a device or a simulator

Step 11: Now, create 3 additional html files that will be hyperlinked through the buttons:

```
stores.html  
maps.html  
ride.html
```

Each of these files should contain at least a header and a footer and should have style that is consistent with the style we defined for index.html.

For now, the content of these pages is not very important, we will continue to develop this application together over the next few labs.

Step 12: (Optional) Finally, we can configure our web app:

We designed our web application to look and behave in a way similar to a native application—for example, it is scaled to fit the entire screen on iOS. We can tailor our web application for Safari on iOS even further, by making it appear like a native application when the user adds it to the Home screen. We do this by using settings for iOS that are ignored by other platforms. ****Note this is optional and can only be tested on an iOS device ****

For example, you can specify an icon for our web application used to represent it when added to the Home screen on iOS by adding the following line to the `<head>`:

```
<link rel="apple-touch-icon"  
href="Dragon_yellow_72x72.png">
```

This is the Share button. You will need to use it in step 12.

Make sure to upload the Dragon_yellow_72x72.png image to the server. Test on an iOS device or simulator by clicking on the logo in



the Safari menu and then selecting “Add to Home Screen”. You should see the Dragon icon on your home screen.

You can also minimize the Safari on iOS user interface by adding the following to the `<head>`:

```
<meta name="apple-mobile-web-app-capable" content="yes" />  
<meta name="apple-mobile-webapp-status-bar-style" content="black" />
```

More on optimizing your web application to Safari:

https://developer.apple.com/library/content/documentation/InternetWeb/Conceptual/SafariVisualEffectsProgGuide/Introduction.html#//apple_ref/doc/uid/TP40008032-CH01-SW1

Use an iOS mobile device (or simulator) to check the solution app at:

<http://www.cs.drexel.edu/~eb452/cs338/Lab1>

Code for index.html

```
<!DOCTYPE html>

<html>
<head>
  <title> BIKE DREXEL </title>

</head>

<body>
  <!-- HTML5 article tag for content -->
  <div id="content">

    <header>BIKE DREXEL</header>

    <!-- Drexel image -->
    

    <!-- H1 means 1st level heading -->
    <h1>Welcome to Bike Drexel</h1>

    <!-- P stands for paragraph -->
    <p>We believe in bicycles! Everything you need to be a Drexel Cyclist.</p>

    <!-- Navigation buttons -->
    <nav id="main_nav">
      <ul>
        <!-- LI stands for list item -->
        <li><a href="stores.html">Bike Stores </a></li>
        <li><a href="maps.html">Bike Maps </a></li>
        <li><a href="ride.html">Go for a Ride</a></li>
      </ul>
    </nav>

    <footer>CS530/338: Developing User Interfaces &copy; CCI, Drexel</footer>
  </div>

</body>
</html>
```