

# CS 1100: Web Development: Client-Side Coding / Fall 2016

## Lab 2: More HTML and CSS

### Goals

- Practice writing HTML
- Add links and images to your web pages
- Apply basic styles to your HTML

This lab is based on what we discussed in class, as well as the readings from this week. The overall goal is for you to become more comfortable with HTML and to explore several ways to style your web pages. We will continue using a model of the Computer Science Department's website as our example.

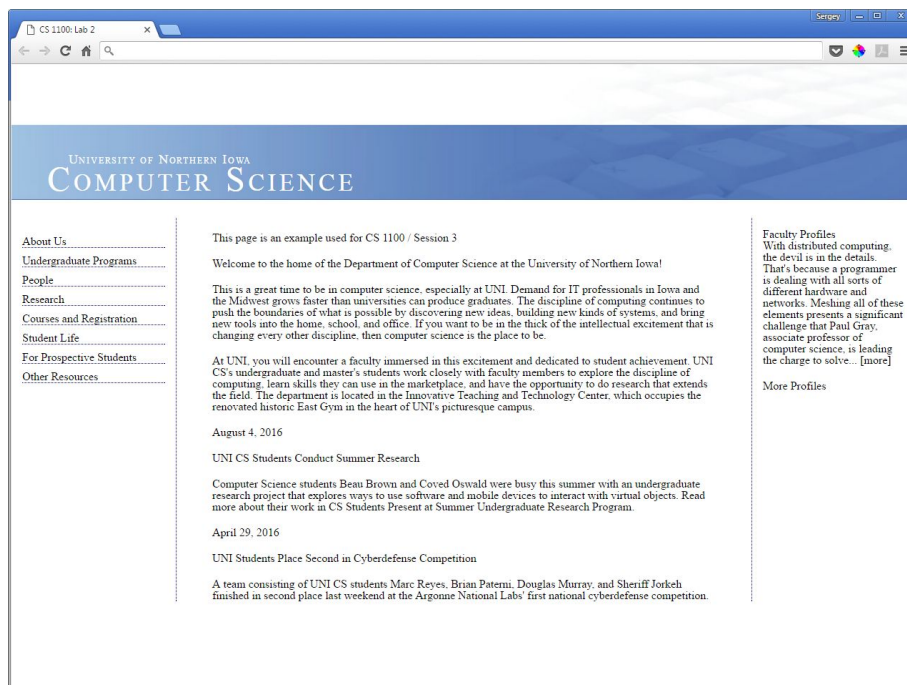
### Step 1. Create your infrastructure

1.1. Create new folder, call it **lab2**.

1.2. Download the file lab2source.zip from the course website and extract its contents into your **lab2** folder. You'll see three files and a folder with 4 image files:

- index.html
- reset.css
- layout.css
- images/
  - gray.jpg
  - project1.jpg
  - project2.png
  - project3.jpg
  - project4.jpg

Open index.html in a browser. You should see the following:



## Step 2. Add HTML

2.1. If you look at the real [cs.uni.edu](http://cs.uni.edu) webpage, you'll find that the text in the center of the page contains 4 inline links:

- ...great time to be in computer science...
- ...historic East Gym...
- ...picturesque campus...
- ...CS Students Present at Summer Undergraduate Research Program...

Open your index.html file in a text editor and make these phrases links as well (use the same URLs). As a reminder, here's the syntax of a link:

```
<a href="link-URL">link-text</a>
```

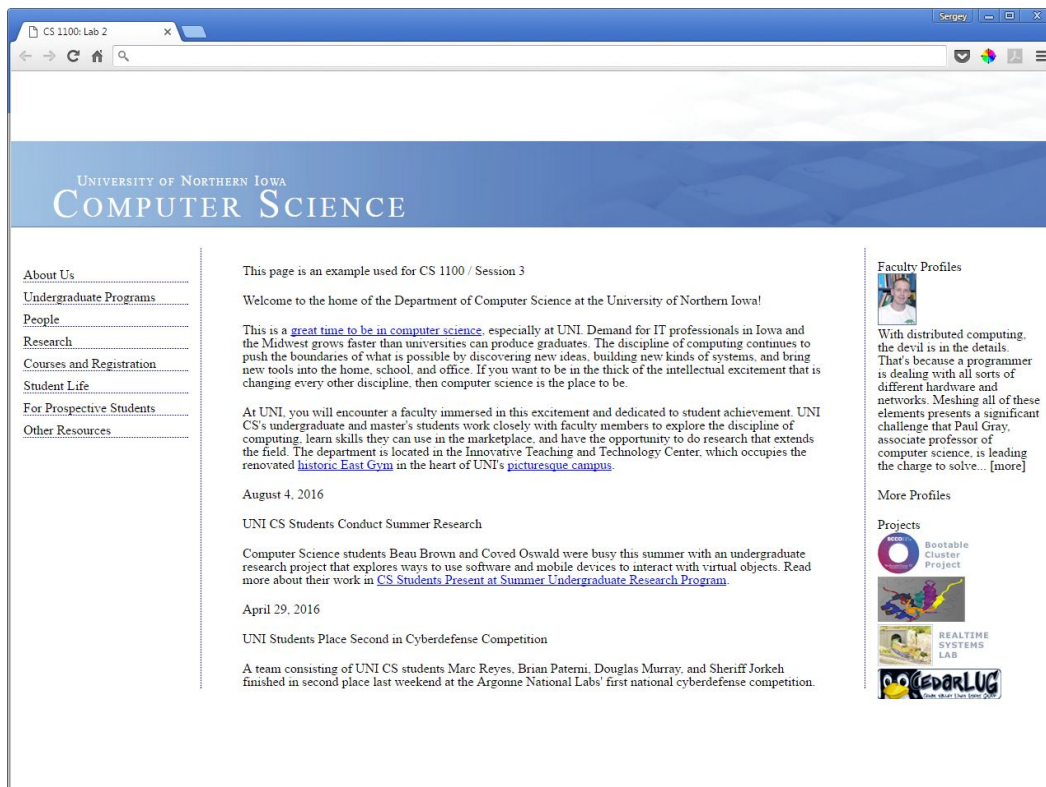
2.2. On the real page, the right sidebar contains 5 images (we'll ignore the top image which is the ITTC building). Your images folder contains the same images. Let's add them to our page. Here's the syntax of an image tag:

```

```

In our case, the path to your image file should include the images folder. For example, the path to the file gray.jpg will be **images/gray.jpg**, so your image tag would be: ****

Now your page looks like this (your links will be purple if you have clicked them):



## Step 3. Add basic styling

**3.1.** Let's change the color of the first link to red! The simplest way to do that is to add a style attribute to the appropriate tag:

before: `<a href="http://cs.uni.edu/why_cs.php">great time to be in computer science</a>`

after: `<a href="http://cs.uni.edu/why_cs.php" style="color:red">great time to be in computer science</a>`

As you may recall, an HTML element may have more than one attribute. In our case, the "a" element (a = anchor) has 2 attributes: "href" and "style. Both are located *in the opening tag only*.

Looks wonderful! Let's add the same color to the remaining 3 links.

**3.2.** It really looks bad. Let's change the color of each link to green instead... WAIT! Don't do it :-)

If you are writing the same code over and over again, you are doing something wrong! In any kind of coding, we try to follow a simple rule: **say it once and only once**. Consider this: we have specified that the color of a link should be red - that should be enough for ALL links on a page. What if there are hundreds of links on a page? What if we want to change it to a different color? Or try dozens of colors before we decide which one to use? So, do the following:

**3.2.1.** Remove all the style attributes from your 4 links.

**3.2.2.** Add the following code into your page's head element (i.e., anywhere between the `<head>` and `</head>` tags):

```
<style>
a {
  color: green;
}
</style>
```

All whitespace is optional; its purpose is readability.

Now we can easily change the color (or any other style property), and our changes will be applied to the entire page.

## Step 4. Add another page

Let's create one more page!

**4.1.** First, let's add two links to the menu on the left:

- "About Us" should become a link to index.html
- "People" should become a link to people.html

**4.2.** Save your file, then make a copy of it and call it **people.html** . Now you have 2 identical web pages.

**4.3.** Open people.html in your editor. Change the contents of the "title" element in the page's head from "CS 1100: Lab 2" to "People".

**4.4.** In people.html, delete the content of the center "div" element: everything between the `<div id="center">` and `</div>` tags.

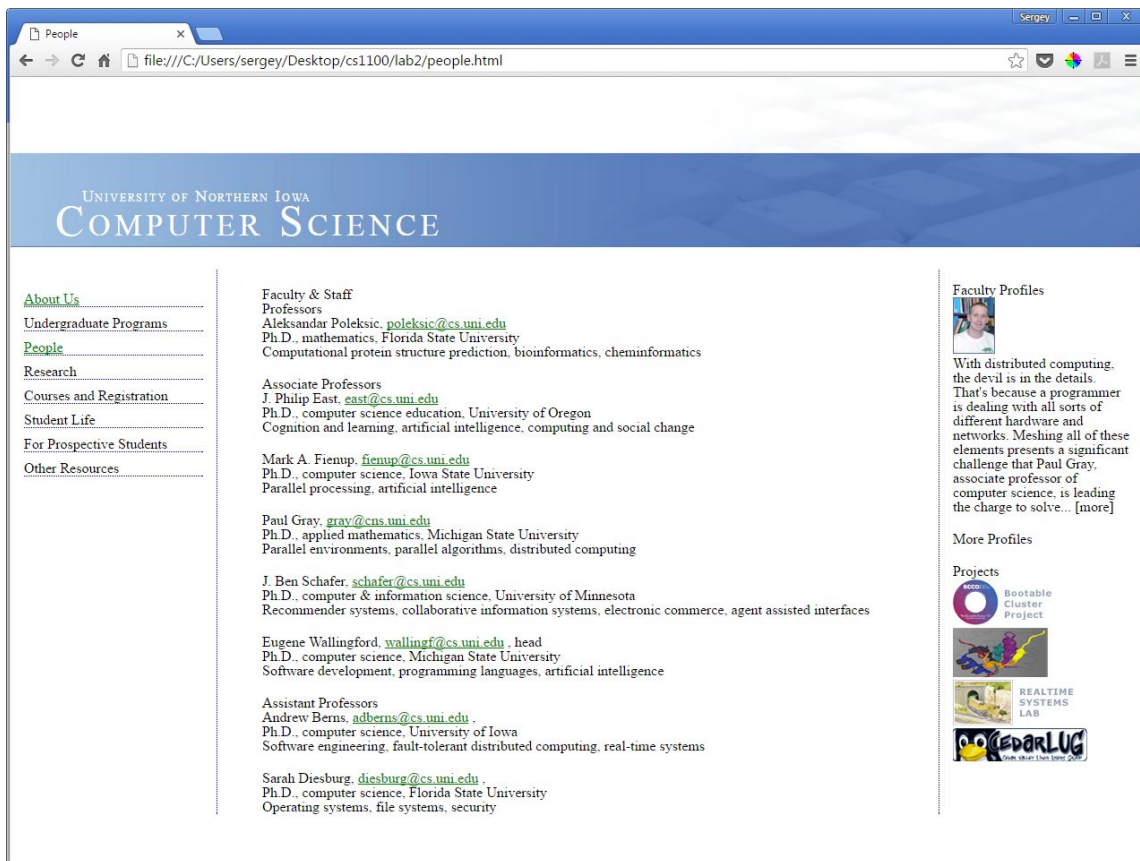
4.5. Add some of the content from the real page: [http://cs.uni.edu/faculty\\_staff.php](http://cs.uni.edu/faculty_staff.php) (Professors, Associate Professors and Assistant Professors):

4.5.1. Copy and paste the text from the web page into your people.html file (into the center "div").

4.5.2. Add the following markup:

- level-1 heading (h1) for "Faculty & Staff"
- level-2 headings (h2) for Professors, Associate Professors and Assistant Professors
- paragraphs (p) for each name
- line breaks (<br>) for education and areas of interest
- add links to each email address: `<a href="">email address</a>` (you may leave the href values empty)

Your page looks like this:



## Step 5. Style your new content

5.1. Let's change the font to a sans-serif font like Verdana.

How do we do that? We could edit the contents of the "style" element on both pages. But what if we had 100 pages? So, again, let's **say it once and only once**:

5.1.1. Create a new text document and call it style.css (the ".css" file extension is required, the name "style" is optional, we use it by convention).

**5.1.2.** Copy the contents of the "style" element into your new file **WITHOUT THE <STYLE></STYLE> TAGS.**

This what should be inside your style.css file:

```
a {  
  color: green;  
}
```

This is the style rule, which says:

*Select all "a" elements and change the color property OF THEIR CONTENT to green.*

**5.1.3.** Remove the "style" element (together with its contents) from **both** pages: we don't need that code anymore.

**5.1.4.** Now we need to link our style.css file (which, by convention, we call a stylesheet) to **EVERY WEB PAGE THAT SHOULD "KNOW ABOUT IT"**. We do that using the "link" element, which is denoted by the <link> tag:

```
<link rel="stylesheet" href="style.css">
```

The value of the "href" attribute is "style.css" - which is the **PATH** to the file containing the stylesheet! (In our case, the file is in the same folder, so the path is just the filename). Place this link inside the "head" element".

Refresh your browser: look at both pages (they are linked from your menu): the links should be green.

**5.1.5.** Now we can change the font! Add the following style rule:

```
body {  
  font-family: Verdana, Arial, sans-serif;  
}
```

This rule says:

*Select the "body" element (there is only one) and change the font-family property OF ITS CONTENT to Verdana. If Verdana is not available, use Arial. If Arial is not available, use the default sans-serif font.*

**5.2.** But now the font is too large! Also, we may want a much smaller font for the right sidebar, a slightly smaller font for the menu on the left, and a medium-sized font for the main content. Essentially, we want to do this:

*Select the left sidebar and change the font size of its contents to a smaller size* (same approach for the other two). So, it would look like this:

```
something {  
  font-size: 0.8em;  
}
```

Look at the HTML in your page: what is unique about the content inside the right sidebar? It's inside a :div" element which has an "id" attribute! The value of the "id" attribute is unique (there can be only ONE element on a page with that ID. So let's use it to select that element!

```
#right {  
  font-size: 0.8em;  
}
```

The "#" character tells the browser that the following word is the value of an "id" attribute, or just an ID. (we will discuss the "em" units later in class; it is equivalent to 80%)

Let's do the same to the other two selections:

```
#left {  
    font-size: 0.9em;  
}
```

```
#center {  
    font-size: 0.9em;  
}
```

Check the page - it looks much better!

**5.3.** Now let's make our headings look like headings:

```
h1 {  
    color: teal;  
    font-size: 1.6em;  
}  
  
h2 {  
    font-size: 1em;  
    font-weight: bold;  
    margin-top: 1em;  
    margin-bottom: 1em;  
}
```

(we'll talk about margins next time)

**5.4.** Last thing: let's modify the way links are displayed: the underlining makes it less readable in the text and in the menu. Let's fix that! Modify the rule for "a" elements in style.css:

```
a {  
    color: green;  
    text-decoration: none;  
}
```

Cool! However, what if we still want all email addresses to be underlined? Here's one way to do it: add a "class" attribute to those elements, give it a name ("email" would be a descriptive name), and then use this attribute as your selector! So:

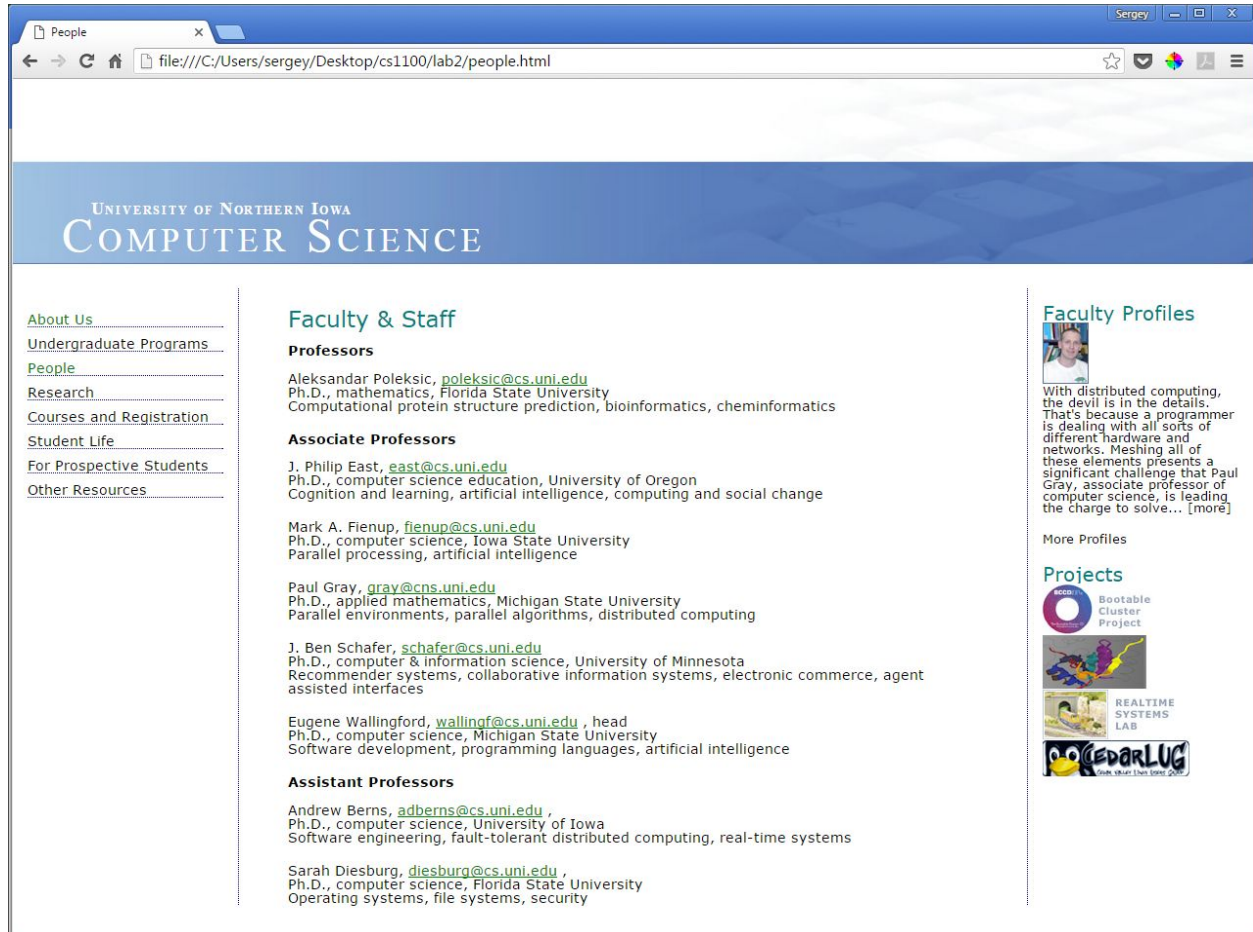
5.4.1. Modify **each** email link (our "href" attributes were empty - which is fine):

```
<a href="" class="email">poleksic@cs.uni.edu</a>
```

Now add this rule to your stylesheet:

```
.email {  
    text-decoration: underline;  
}
```

And here's the result:



## Submit your work

Save your lab1 folder as a zip file and submit the zip file to eLearning:

<https://bb9.uni.edu> > log in with CatID > our course > Course Content > Labs > Lab 2

(make sure you don't submit the lab2source.zip file instead)