EECS 448
Software Engineering Lab
Lab #8: Web Languages Intro

Due Time

- This lab is due Nov 8,2020 11:59 PM.

Lab Type:

This lab is an individual lab.

Additional Resources

W3Schools: Great interactive resource for all web languages

Topics

1) HTML

2) CSS

3) JavaScript

Overview:

HTML

HTML: html, body, and head

All HTML documents are surrounded by the <html> tag. It is an open and close tag that then contains the <head> and <body> tags. The head contains some of the behind the scenes mechanics. One tag you can put in the head right away is the <title> tag. The title tag controls what a browser tab or window title bar's text is.

The body contains all the content that the is contained with the web page. Every, image, link, paragraph, or table resides here.

```
<html>
        <head>
                <title> My first web page </title>
        </head>

        <body>
        In this sentence I have
        <br>
        <u>underlined words</u>, <i>italicized words</i>,
        <br>
        and <b>bold words</b>.
        </body>
```

```
</html>
```

HTML: Images

You can put images in your web page, but first you need to pick an image to embed. There are two ways to refer to an image:

- Download and use a local image
- Refer to the address of an existing image

```
<img src="theFileNameHere.jpg">
```

The img tag is a stand alone tag like the break tag. It doesn't surround anything.

Notice how we added the src="theFileNameHere.jpg" portion? That's because the tag alone isn't enough information. We can't just tell the browser we need an image, we need to tell the browser which image we want. If we just list the file name, the browser will assume it's in a local folder. What you added was an HTML attribute. All attributes have a similar format:

<tag attribute = "value">

You pick some attribute then set it equal to a value in quotation. There are many attributes and values for various tasks. For example, depending on your image, it may be very large. You can control that. You just need to add a height and a width attribute.

```
<img src="theFileNameHere.jpg" height="300" width="500">
```

You can control the pixel heights and widths of the image. You can also do percentages if you like:

```
<img src="theFileNameHere.jpg" height="20%" width="50%">
```

The other way we can put an image in our page is link to an existing one instead of downloading a copy. To do this, you can go back to that image you found and instead of saving a copy, choose "copy image location" (or the wording that matches your browser). You would then replace the file name with this URL. Sometimes these URLs get very long, so you don't have to do this right now, but it's good to be aware of.

Reference: http://www.w3schools.com/tags/tag_img.asp

HTML: Anchor tags

To create clickable links, you will use the anchor tag, or <a> for short. Here's the template:

```
<a href="http://completeURLhere.com"> The words you want to be clickable </a>
```

For example if I wanted to make a link that took people to google when they clicked on the words "Go to google" I would do the following:

```
<a href="http://google.com"> Go to google </a>
```

Remember to include the "http://" part. Even though you don't type that in browsers anymore, you'll need it in your code. It acts as a flag that tells the browser it need to use the Hyper Text Transfer Protocol, or in other words, it needs to grab something from the internet instead of something local.

Reference: http://www.w3schools.com/html/html_links.asp

HTML paragraphs

Paragraphs are used to create, well, paragraphs. The text get some automatic formatting, like ensuring its not a the same line as some other piece of content. The template is:

Your paragraph here

Reference: http://www.w3schools.com/html/html_paragraphs.asp

Javascript

JavaScript is language that runs client-side and is used to make your HTML pages interactive and responsive. It is not a scripted version of Java, it's a whole other language.

A lot of syntax is the same or similar to C++/Java:

if statement

boolean operators

loops

functions (mostly)

Important changes:

- All variables are declared with the key words *var*
- There is no explicit class key word (see reference below)
- JS files don't include each other, instead they are all included in your HTML document

Useful reference and tutorials on W3Schools.

Connecting JavaScript to an HTML file

In the head tag of your HTML file, include the script tag:

```
<html>
  <head>
    <script type="text/javascript"
        src="fileName.js">
    </script>
  </head>
```

The script tag is and open-close tag that we can put our JS code in, but it helps modularity to simply link to another file.

Events

JavaScript functions can be called in reaction to an event occuring. For example, we can define a button in HTML to call a JavaScript method.

```
<!-- In html -->
<button onclick="myFunc()">
//In JS file
function myFunc()
{
   alert("You called my function!");
}
```

Other useful events:

- onclick
  - when a tag is clicked
- onmouseover
  - when the mouse browses over a tag
- onmouseout
  - when the mouse leave the area of a tag
- onload
  - when a tag is loaded
  - setting this event in the body tag will call a JavaScript function when the page loads

Manipulating HTML

JavaScript can manipulate nearly every aspect of an HTML dynamically. To give JavaScript control over an HTML element you must do two steps:

- In the HTML file
  → Give a tag you want JavaScript to be able to affect an id attribute
- In the JavaScript file
  → Find the element by it's id
  → Manipulate the attribute in question

```
<!-- In HTML -->
<img src="file.jpg" id="myImgId" width="100" height="200">
<br>
<button value="ZOOM!" onclick="zoomIn()">
//In JS
function zoomIn()
{
  var theImgTag = document.getElementById("myImgId");
  theImgTag.width = 1000;
  theImgTag.height = 2000;
```

```
}
```

Exercise 1: Password validator

Create a webpage that allows the user to enter a password two times in order to validate it.

Web page content:

- Two password fields
    - first to enter the password and a second to verify it
- A button labelled "Validate" that alerts one of the following messages:
    - Display informative error message if any of the following occur:
        - the passwords entered don't match
        - the passwords are not at least 8 characters long

Exercise 2: Slide show!

Create a new webpage that has a single img tag and two buttons labeled previous and next. Your slideshow should.

- contain at least five pictures of your choice (appropriate for class)
- cycle through all the pictures
- wrap around to the beginning if I keep pushing next
- wrap around to the end if I keep pushing previous
- force the images to be all the same size regardless of the original image files' dimensions

CSS

CSS: Introduction

CSS, or Cascading Style Sheet, is a whole other language from HTML (that's right, you're two languages in one day). It's entirely a helper language used to define how HTML should appear. It includes styling details such as, colors, sizes, font families, and background image.

CSS: Setup

In the same folder your webpage is in, create file called myStyle.css. Note the different extension? That's because this file will only contain CSS code. Now add the following code to your CSS file:

```
p
{
   background-color: blue;
}
```

Now, go back to your HTML sheet. Inside the head tag, put the following:

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

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

Now make sure all the files are saved and refresh your browser. You should see your paragraph now has a blue background.

CSS: Basic

All styling code follows this template:

```
tagName
{
  style-attribute1: someValue;
  style-attribute2: someValue;
}

someOtherTag
{
  style-attribute1: someValue;
  style-attribute2: someValue;
}
```

You can style almost any HTML tag. There are many attributes, such as background-color, and values, such as, blue. We'll discuss a few here.

CSS: Background colors

You already have a background color for your paragraph. Now let's give one to the whole page! To do so, we'll style the tag that surrounds everything, the body.

```
body
{
  background-color: red;
}
```

There are lots of color keywords, such as red and blue. There are many others, but if you need a very specific color, you can use W3Schools' color picker.

Reference: http://www.w3schools.com/tags/ref_colorpicker.asp

CSS: text color

You can give the text in your web page it's own color. The "color" attribute colors the color the text inside of a given tag.

Example:

```
p
{
   background-color: red;
   color: white;
}
```

Now all paragraphs will have a red background and white text.

CSS: Font

There are many things we can change about font. The color attribute controls the color of text, but there are several attributes that apart of the font attribute collection. Here's an example:

```
p
{
   background-color: red;
   color: white;
   font-family: arial
   font-size: 22pt;
}
```

Now all paragraphs will have 22pt (the size), white arial font on a red background. Reference:

Valid fonts: http://www.w3schools.com/css/css_font.asp

More about CSS font attribute: http://www.w3schools.com/cssref/default.asp#font

CSS: background images

Lots of tags can have a background image. We'll give our web page a particular background image. You could give your lists or paragraphs background images too, but they probably aren't large enough to utilize them well.

The following code give the body a background image of a gif (assuming it is in the same folder) that will stretch to cover 100% of the screen (the size) and not repeat.

```
body
{
   background-image: url(img_flwr.gif);
   background-size: 100%;
   background-repeat: no-repeat;
}
```

You can also play with how the background image behaves when scrolling by tinkering with the position. For example if you don't want is to move when the user scrolls do the following:

```
body
{
background-image: url(img_flwr.gif);
background-size: 100%;
background-repeat: no-repeat;
background-attachment: fixed;
}
```

Reference: Basic: http://www.w3schools.com/cssref/pr_background-image.asp

Position: http://www.w3schools.com/cssref/pr_background-position.asp

Size: http://www.w3schools.com/cssref/css3_pr_background-size.asp

Repeat: http://www.w3schools.com/cssref/pr_background-repeat.asp

CSS: hover

CSS can also detect some simple events that occur on your page and adjust the styling accordingly. For example, you can change the styling of a tag to resond when a user has their mouse over it.

Let's look at two blocks of styling for the anchor tag (our clickable links):

```
a
{
   background-color: blue;
   color: red;
}

a:hover
{
   background-color: red;
   color: blue
}
```

With these two blocks, the user will see red text with a blue background for all links normally, but when they hover over the link, the styling will invert. Now, you can adjust any styling you want. You could make a paragraph that increases its font size when hovered over or a list that changes from arial to courier font when hovered over.

Reference: http://www.w3schools.com/cssref/sel_hover.asp

CSS: classes

Sometimes you don't want all tags to receive the same styling. One option is to create classes. It's a two part process. On the CSS side you need to create a styling block that will contain the informatin for your special tags.

```
p
{
   color: blue;
   background-color: yellow;
}

p.halloween
{
   color: orange;
   background-color: black;
}
```

Default paragraphs will have the top block of styling, but any paragraph that I choose to can have the halloween themed styling. I just have to tell HTML which ones I want to be special. To do so, I include the class attribute on the HTML side:

```
<p> Normal paragraph </p>
<p class=”halloween”> Booo! </p>
```

Exercise 3: Personal Profile

Make a profile page that is styled with CSS. Have fun with this. You can make a profile about anybody or anything. You are not required to make this about yourself or post any personal information online. This profile can be entirely fictional.

HTML Requirements:

- Profile picture
- Biographical paragraph
- Hyperlinks to favorite web sites
- Embeded youtube video

CSS Requirements:

- Use all the attributes listed in the CSS section at least once.
- The page should not induce seizures or headaches - in other words, mind the color pallet you choose
- All text should be easy to read

CSS manipulation through JavaScript

The Style Attribute

JavaScript, in addition to accessing and manipulating HTML, can change the styling of any tag you give it access to. The reality is that there is a style attribute that nearly every tag has. For example, one could style a paragraph tag using the style attribute:

```
<p style="background-color:red"> This text will have a red background </p>
```

Levels of Styling

The previous example was inline level styling. The other levels of styling, in order of precedence, are:

- inline: the style code is given in the tag
    - highest precedence, but only applies styling to that one tag
- document level: the style code is written inside a style tag in the head of the HTML document
    - second precendence, but only applies to the document it's written in
- external: what we do in this lab, writing the code in an external CSS sheet
    - lowest precedence, but the most modular

The JavaScript style Object

The **D**ocument **O**bject **M**odel, or DOM, that defines the structure of an HTML document has inside of it a style object. To access the style object, you simple need to access the tag, via its id, like in previous exercises. You can then access the style object of that tag.

Exercise 4: CSS Manipulation

Create a web page that have a paragraph with some dummy text. Near the paragraph have text fields to accept the following values:

Border:

- red value
- green value
- blue value
- width

Background color:

- red value
- green value
- blue value
- width

Finally, have a button that, when clicked, changes the border and background color to be as specified. You can use either the rgb() method or a color code, but you should tell the user what units they are in (00-FF or 0-255).

```
//in JavaScript


//Access the tag:
var someTag = docuement.getElementById("theTagsId");
```

```
//Change the style attribute
someTag.style.backgroundColor = "red";
```

NOTE: the names of the style attributes in CSS are all hyphenated (e.g. background-color) but in JavaScript they are nearly all converted to camel-case.

Here is a reference of the style properties that JavaScript can access and change.

Excercise 5: Publication

In addition to uploading your code to github, place your files in the public_html on your EECS account. You can then access them through the URL:

```
http://people.eecs.ku.edu/~yourKUuserNameHere

Example:
http://people.eecs.ku.edu/~jrmiller
//Display Dr. Miller's EECS homepage
//Why not mine? Mine is a hot mess.
```

Making an index.html

When you go to your people.eecs page, you'll most likely see a simple file listing. This is the default behavior of apache. To make a landing page, you can write an **index.html** file.

In your index.html

- Use links, JS, or whatever means you like to create a menu to all your other exercises
- Spruce it up with styling so it's more than just four hyper links - apache could have given us that without any effort

If you already are using your default homepage just provide us with a sub-folder URL.

Example:

```
http://people.eecs.ku.edu/~KuUserName/subdir/eecs448/lab03
```

Note:

There have been reports of some students' public_html folder not having the correct permissions set to be publicly accessible. IT gave me a shell script that you can copy and paste (yes, you can copy and paste this) into a .sh file then run it in your terminal.

Script:

```
#!/bin/bash
chmod 751 ~
chmod 755 ~/public_html
find ~/public_html -type d -exec chmod 755 {} \;
find ~/public_html -type f -exec chmod 644 {} \;
sh ./scriptName.sh
```

You can also see the commands you could just run from terminal and type them manually.

Web language Summary

"Are all these webpages I look at everyday just text files in directories with code written in them?"

Yep.

Sorry kid, the world wide web isn't magic. It's science. Computer science to be exact.

Rubric:

- [20pts] Exercise 1: Password validator
- [30pts] Exercise 2: Slide show
- [20pts] Exercise 3: Profile
- [20pts] Exercise 4: CSS Manipulation
- [10pts] Exercise 5: Publication

Submission

Github URL

people.eecs.ku.edu/~ URL