Home Print Font↑ Font↓

CSCI 341 : HANDS ON

UNIT: IDed & Styled

Activities

- Recognize **the three layers** of a web page.
- Create a simple web page.
- Include two html elements with text that displays on the screen.
- Give each element its own unique id.
- Use the id to explore styling with css.

Overview

Computers create models of the world using data. This semester, we are using a pirate theme for our labs, so for this lab we will create a data model of two pirate ships in html. The "pirate ships" will really be html elements. The following examples will use the <div> element, but feel free to use <p>, <span>, or any other element you like. Whichever element you choose, please ensure that it is standard for HTML5. In this class, we do not want to use deprecated elements.

> *Deprecated elements* are those elements that were once standard but have been replaced by an improved alternative feature. So, they still work just to keep the older web pages working. However, they should not be used in new software. It would be like coming to class dressed like The Fresh Prince of Bel-Air. It was once acceptable and could still work most of the time, but there are social consequences. Now, there is a better way. Word.

After creating a simple web page with at least two elements, you will give your elements unique names. Then, you will use those names to explore some styling with css. That sounds exciting, so let's get started by creating a simple web page.

Activity One - A Simple Web Page

To create a very simple web page, just follow the steps below.

1. Follow this link to view a very **simple web page**.
2. View the source code for the page.
3. Copy the source code and paste it into a new file.
4. Save the file to your local computer where you can work with it. You can give it any name you like, but I called mine "simple.html".
5. From your editor, update the header block author's name and date to make it yours.
6. Change the page title to something that is meaningful to you. That is the text between the opening and closing <title> tags. It appears on the tab in your browser.
7. Save the file again.
8. Test the page by loading it into your browser.
9. If it works, upload it to the server in the same folder with the html file from your **Web Page Redux** lab.
10. Test the page again by loading the copy stored on the server into your browser.
11. Run the code through the **validator** by pasting the url of your new page into the address box on their form and clicking the button that says "check".
12. If that goes through without errors, you are ready for the next step!

Activity Two - adding a couple of elements with ids

Include a <div> element between the opening and closing <body> tags in your web page:

```
<body>
  <div>This text will appear on the page.</div>
</body>
```

The <div> element defines a section in your document and has an opening tag (<div>) and a closing tag(</div>). The text between the tags will appear on the page.

Sometimes, an element needs to be given a unique id. The id can be used to define styling in css or to add functionality to the page using JavaScript. To give your element an id, use the id attribute.

```
<body>
  <div id="fred">Yabba Dabba Doo!!!</div>
  <div id="barney">Hiya, Fred!</div>
</body>
```

In the example above, the <div> called "fred" will display "Yabba Dabba Doo!!!" on the page, and the <div> called "barney" will display "Hiya, Fred!" When loaded into the browser, it will look like:

```
Yabba Dabba Doo!!!
Hiya, Fred!
```

The id of each element can be used to distinguish one from the other - like calling a person by their name. Each id attribute on a page must have a unique name. So, you cannot have two elements where the id="fred". A programmer can use an element's id to create a styling rule using css, and that just happens to be our next task!

Activity Three - css styling

A css rule set consists of a selector and a definition block. Here, we will select a <div> element by its id. A selection by id begins with a hash character (#) followed by the id name.

```
#fred {
  color: orange;
}
```

The definition block is defined by the curly braces, and each styling rule ends with a semicolon. In this example, we are defining the color of the text to be orange. You might be wondering how to link this css rule to the html. Just follow these steps for our Flintstone example:

1. Create a new file called "simple.css" with the above styling definition.
2. Save the file in the same folder with your html file.
3. To link the css file to the html file, include the following line in the <head> section of your html.

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

   If you copied "simple.html" to start your page, it should already be there. This line tells "simple.html" to link to "simple.css" to get the styling information.
4. Reload simple.html and see if the colors change. If it worked, "Yabba Dabba Doo!!!" will appear in orange.

Next, we will give some styling to barney:

```
#fred {
  color: orange;
}

#barney {
  color: brown;
}
```

Adding a selection by id="barney" allows us to color barney's text brown. After adding the selection for barney to your css file, save it and reload your html page. You should see "Yabba Dabba Doo!!!" in orange and "Hiya, Fred!" in brown. When you reload your page, it should look remarkably similar to **flintstone.html**. It might be tempting to just copy and paste the sample code, but it will help you to learn the details of the material presented here if you type it yourself.

That was fun! If you've followed the example, you should have a very simple page with two elements that are styled with CSS. It is time for you to go out on your own and create your first pirate-themed page.

Lab Instructions

This is the part where you **create your own page. Do not turn in the Flintstones example above**. That example is a teaching exercise and is not the IDed & Styled Assignment. For the real assignment, we are using a pirate theme. For this lab, your elements will represent pirate ships.

1. Create an HTML page, and give it a piratey-sounding name.
2. Include at least two elements on your page. Feel free to use <div>, <p>, <span>, or any other element that meets your needs.
3. Be sure to use both the opening and closing tags on your elements.
4. The text of each "pirate ship" element should contain the ship's name, the captain's name, and a description of the ship's mission.
5. Give your elements unique pirate ship names for their ids. Remember that they represent pirate ships, so their names should sound like they came from pirates. **You'll walk the plank if your ship names are "fred" and "barney"!**
6. Create a simple css file. The root name of the css file should be the same as your html file. For example, if your html file is called "flintstone.html" (but you wouldn't do that because the Flintstones are not pirates), your css file should be called "flintstone.css".

7. Your css file should use your pirate ship ids to style the text of your pirate ship. Use styling to change the color, font, or font-weight of each pirate ship to something creative. If you want to make it really special, you can use a background image to add color to your page. Check out **w3schools** for cool ideas.
8. Be sure to run your code through the **validator** to check it for errors. This is also a quick way to diagnose problems if your page is not working.
9. Check through the rubric for IDed & Styled in Canvas Assignments to ensure that you have earned all possible points.
10. Add a link called "IDed and Styled" to your personal n341 course page.
11. Submit your lab in Canvas for grading.

Important Procedures for All Labs

Here are some general notes for perfection that you should follow for every assignment:

1. Please produce all web content to HTML5 standards.
2. Please **validate** all your files.
3. Be sure to update the header block comments for each file.
4. Be sure to check your browser's console / developer tools for error free code.
5. Test your code in Chrome and Edge at a minimum.
6. Use only your own original code for all labs.
7. Be sure to put your CSS and JavaScript in a separate files from your html.
8. Be sure to read through the lab rubric in Canvas.
9. Submit your lab in Canvas for grading.

Holler if you have any questions!

Mission Accomplished!

Now that your pirate ship has a name, you're leaving port with style! Yo Ho Ho!