CSCI 341 : HANDS ON

UNIT: Multi-Writes

Activities

- **Explore three different ways to send output** using JavaScript.
- Create a page that outputs using **textContent** to one page element.
- On the same page, output using **innerHTML** to a second page element.
- Create a page that outputs using **document.write**. For an extra challenge, try to get all output to be visible simultaneously on the same page.
- Post assignment as a link from your personal index page.
- Submit link in Canvas for grading.

Overview

In this lab, we will explore three different ways to send output (or write) to a web page using JavaScript.

1. .textContent
2. .innerHTML
3. document.write()

Before beginning this lab, click the orange link above to see a simple demonstration of JavaScript output.

Activity One - .textContent

One way to send output to a web page is to use the textContent property. It allows you to either set or retrieve the text from a selected element. On retrieving text, textContent returns the text from inside the element. The following html and JavaScript shows how to **retrieve** text.

```
<!-- Sample html fragment -->
<body>
   <div id=″myFavElementId″>JavaScript is awesome!</div>
</body>
```

In the code example below, a string variable called strMessage is declared and set to null (nothing). Next, the element variable elOutput is declared and set to hold the location of the html element with the id=myFavElementId from the html above. This lets Javascript know to access the <div> element.

On the fourth line, strMessage is set to the text that is contained in the html element.

```
// Sample JavaScript fragment
var strMessage = ″″;
var elOutput = document.getElementById(″myFavElementId″);
strMessage = elOutput.textContent;
```

When strMessage is set to the textContent of elOutput, its value is "JavaScript is awesome!" This value is taken from the html.

The textContent property can also be used to **set** the text value on the html page. On setting text, this property removes all of the text from within the element and replaces it with the set value. The following is an example of setting text.

```
var strMessage = ″Text we are writing to the page.″;
var elOutput = document.getElementById(″myFavElementId″);
elOutput.textContent = strMessage;
```

The first line declares a string called strMessage and sets its value to "Text we are writing to the page." Next, elOutput is declared and set to hold the location of the html element with id=myFavElementId. The third line sets the textContent of the element to the value of strMessage.

So, after the JavaScript executes, the text on the page will say "Text we are writing to the page." instead of "JavaScript is awesome!"

See page 64 of the textbook for another example of textContent in use. Then, create a JavaScript page that outputs text declaring your major in pirate talk, using textContent. For example, "I be from across t' computer science sea."

Activity Two - .innerHTML

The innerHTML property works much like the textContent property. The difference is that it can set or return the HTML content (inner HTML) of a selected element. Not only can it retrieve or set the text of an element, but it can also include html.

```
var strMessage = ″Text we are <strong>writing</strong> to the page.″;
var elOutput = document.getElementById(″myFavElementId″);
elOutput.innerHTML = strMessage;
```

In the example above, we can include some simple formatting. The html <strong> tag is used to define important text - usually by making it bold. So, the word "writing" would appear on the web page in bold.

See page 65 of the textbook for another example of innerHTML in use. Then create a Javascript page that outputs text informing us of one fact about yourself in pirate talk, using innerHTML. For example, "I enjoy a good sea chantry with me first mate."

Activity Three - document.write()

The keyword "document" is an object which represents the entire web page. Through dot name nomenclature, you can access the document object's write() method. Dot name nomenclature simply means that we place a dot operator between the object name and its method. The write() method accepts a single string parameter which is written to the page.

The write() method is mostly used for testing. Please keep in mind that it will **delete all existing HTML** if it is used after the document is fully loaded. So, putting it inside a document.ready jQuery wrapper will cause your written message to be the only thing on the page.

```
document.write('Good Afternoon!');
```

The example above is from page 50 of the textbook. Have a quick look at that page for a detailed explanation of the document.write() method.

Then create a JavaScript page that outputs text using document.write() to introduce yourself in pirate talk. For example, "My name be Captain Michele Roberts. Avast ye scurvey dawgs!"

Lab Instructions

1. Create an HTML5 page that has at least two page elements for output and JavaScript file .
2. Use JavaScript to produce output to one page element using textContent.
3. Produce output to a second page element using innerHTML.
4. Produce output using document.write().
5. For an extra credit point, produce output for document.write() on the same page as the other two. You must be able to see output from all three methods on one page at the same time.

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!

Arg! Hoist the colours. The wind is in our sails. Enjoy some loot and its off to the next adventure!