**I. Your Name:**

**II. Instructions:**

1. Change this file’s name to **lastName-firstInitial-3B-quiz.docx**
2. Complete the questions below by typing your answers into this document
3. This will be graded and counts as a “Checkoff”
4. This is open Internet, open brain, and open VSCode. Feel free to test out your JS answers by running them in a web browser
5. When you are done, go ahead and ZIP up this file and post it to the dropbox before it closes

1) Write code that:

A) turns the colors string (below) into an array

B) now loop through this array of colors using a for..of loop and

C) create an HTML ordered list (it will be a String)

D) Add this list to the HTML page inside of an existing <div id="output"></div>

const colors = "red,green,blue";

const colorArray = colors.split(",");

let html = "<ol>";

for (let element of colorArray)

{

html += `<li>${element}</li>`;

}

html += "</ol>";

document.querySelector("#output").innerHTML = html;

2) Now do the same thing, again, as question #1 above, but this time utilize array.map() and array.join() (instead of a for..of and string concatenation)

const colors = "red,green,blue";

const colorArray = colors.split(",");

let html = `<ol>${colorArray.map(element => `<li>${element}</li>`).join("")}</ol>`;

document.querySelector("#output").innerHTML = html;

3) Encode the following information into a *well-formed* XML file named **app-data.xml**

* the title of the app (which, for example, might be used in a <title> or <h1> tag) is “Greeter"
* a list of greetings - "Hi", "Hello" and "Hey there”
* The *root element* must be named <greeter-data>
  + You can name the other elements whatever you wish
* Make sure that you follow the other rules of XML (see the **ajax-2** assignment linked from **PE-03**)
* Recall that (if you want to) you can always verify that an XML file is *well-formed* by opening it in a web browser

**app-data.xml**

<?xml version="1.0" encoding="UTF-8"?>

<greeter-data>

<title>Greeter</title>

<greetlist cid="greetings">Hi, Hello, Hey there</greetlist>

</greeter-data>

4) Now encode the same information from the above question into a well-formed JSON file named **app-data.json**

* Recall that (if you want to) you can always verify that a JSON file is well-formed by heading to <https://jsonlint.com/>

**app-data.json**

{

"greeter-data": {

"title": "Greeter",

"greetings": ["Hi", "Hello", "Hey there"]

}

}

5) In an XHR callback function for onload or onerror, we often have a parameter named e or evt, which holds an Event object. In this context, what does e.target represent?

e.target represents the element that triggered the event e. In this context that represents the XHR object. We then use this to get access to the XHR’s response text by calling e.target.responseText.