So far we have used Python to get you familiar with basic coding concepts, such as variables and assignments. Today we will be using JavaScript to apply the same knowledge to create simple GUI messaging.

A. JavaScript in HTML

We’ve now spent a couple of weeks learning HTML and how to use it to create basic web pages.

I. HTML Pros:
   a. Good for displaying information (like a brochure)
   b. Easy to use

II. HTML Cons:
   a. Can’t perform calculations
   b. Can’t really interact with the user

JavaScript fixes most of these issues, and in this lab we will outline many of the concepts behind JavaScript to make your webpages even better.

Step 1:
Create a new folder named Lab7 in your CSE3 folder. Open up Notepad++ and save your file as SuperBowl.html

Step 2:
Create a basic HTML skeleton. Please refer to the other labs if you do not remember how to write the HTML skeleton. Put a pair of `<center>` tags as the first thing in your `<body>` tags. In order to use JavaScript in your HTML, you have to put all of that code in between a pair of script tags. Place these tags inside of your `<body>` tags (inside the `<center>` tag of the `<body>` tag):

```html
<script type= "text/javascript"> </script>
```

Step 3:
Inside of the script tags, we are going to be writing JavaScript code, not HTML code. We will start by creating some variables and practicing with JavaScript so that you can familiarize yourself with it before creating the webpage for your lab.

Variables in JavaScript, for the most part, operate similarly to variables in Python. They can be assigned a numeric, string, or Boolean value and each unique variable can hold one value at a time. However, there are two key differences with how variables are handled in JavaScript:

1. To declare a variable in JavaScript, you **MUST** have the keyword `var` before your variable name.
2. All variable assignments/declarations **MUST** end with a semi-colon. As you will see in later examples, the same applies for almost every code you write in JavaScript.
Below are examples of legal variable use in JavaScript:

```javascript
var object;  //Legal declaration of variable "object"
var name = "John";  //Legal string assignment to variable "name"
var age = 21;  //Legal int assignment to variable "age"
var isWeekday = true;  //Legal boolean assignment to variable "isWeekday"
```

Below are some illegal variable usage:

```javascript
var item  // Illegal declaration - Missing semi-colon
width;  // Legal but not recoment - Missing "var" identifier
length = 5;  // Legal but not recoment - Missing "var" identifier
var height = 9;  // Legal int assignment to variable "height"
var height = 10;  // Do not try to re-declare the same variable name
```

**Step 4:**
Let’s use JavaScript to calculate how old you are.

I. Create 3 variables: **name, yearBorn, age**
   a. You can have each variable declared on its own line with its own keyword `var` like so:
   ```javascript
   var name;
   var yearBorn;
   var age;
   ```
   b. Or, you can combine them on one line and separate them with commas with only one keyword `var` and one semicolon, like so:
   ```javascript
   var name, yearBorn, age;
   ```

II. Now we can assign values to **name** and **yearBorn**. Go ahead and put down your own name and the year that you were born.
   a. **name** is of type string, so we have to put its value inside quotation marks
   ```javascript
   name = "Henry";
   ```
   b. **yearBorn** is just a number, so NO quotation marks are needed
   ```javascript
   yearBorn = 1996;
   ```

III. With these new values, we can now calculate the value of age:

```javascript
name = "Henry";
yearBorn = 1996;
age = 2016 - yearBorn;
```
**Step 5:**
Now that we have all of our information, we need to display it somehow. In JavaScript, there are **TWO** ways to display info on a webpage:

I. `document.write();`
II. `alert();`

document.write() spits out information directly onto the webpage, whereas alert() shows the information in a pop-up box. Use document.write() to write our name and alert() to write our age:

```javascript
document.write("My name is " + name);
alert.age();
```

**B. Operator Overloading**
We all know what the plus sign (+) does right? It just adds two numbers together. But, in JavaScript, it can do much more. There are so many things we want computers to do and so few symbols on the keyboard, so to overcome this, we sometimes **overload** a symbol by having it do different things in different situations.

I. A plus sign can:
   a. **Add two numbers together**
      i. $5 + 5 = 10$
   b. **Concatenates two strings together** (think of concatenating as stapling 2 strings together to get 1 bigger string)
      i. “Super” + “Bowl” = “Super Bowl”
      
      **NOTICE** the space after the letter ‘r’ in “Super”. If you had “Super” + “Bowl”, then JavaScript would do “Super” + “Bowl” = “SuperBowl”.
   c. **Concatenates a string and a number together** (when it sees a string and a plus sign, JavaScript will just slam everything together):
      i. “Super ” + 8 = “Super 8”
      ii. “Super ” + 47 = “Super 47”
      iii. 6.7 + “Super” = “6.7Super”

If there are more than 2 items in the expression, follow the order of operations:

```
5.5 + 5.5 + “Super” = “11Super”
5.5 + (7.4 + “Super”) = “5.57.4Super”
“Super” + 5.5 + 7.4 = “Super5.57.4”
```

**Step 1:**
Use concatenation to create a sentence with your name and age in it that you display using document.write()

I. Texts that are contained within quotation marks are considered to be **literals**. This means that whatever is inside the quotations is the exact text to be printed by the webpage.
II. name and age are variables in the sentence, so they must go OUTSIDE the quotation marks. This way, JavaScript knows to look for their value instead of just printing the texts “name” and “age”.

```javascript
document.write("My name is " + name + ". I am " + age + " years old.");
```

C. Interacting with the program
So far we have learned how to use variables, calculate values, and display information to the user. We haven’t yet shown how to INTERACT with the program.

For now, the easiest way to get information from the user while the program is running is to use the **prompt** function. We can ask the user what his/her name is like this:

```
name = prompt("What is your name?");
```

Try it and see what happens!

![JavaScript prompt](image)

**Step 1:**
Modify your program to ask for the user’s name and yearBorn using **prompt();**

```
name = prompt("What is your name?");
yearBorn = prompt("In what year were you born?");
```

When you run the program, the user should be prompted to input their name and the year that they were born in. Once they do that, the webpage should print the appropriate sentence containing the user’s name and age.

D. Conditional Statements
Conditional statements allow you to make choices and do different things with your code depending on the situation. You can think of conditional statements as acting like branches in a flow chart or decision tree.

Remember **if statements** from Python? JavaScript (and all other real programming languages) also contains **if statements** and they function pretty much the same way.

As a review, **if statement** checks if the part inside the parentheses (called the **condition**) is true.
I. If the condition is **TRUE**, then the code inside the body is **EXECUTED**.
II. If the condition is **FALSE**, the code inside the body is **SKIPPED**

**NOTE**: Unlike Python, JavaScript utilizes **curly braces** to indicate the body of the statement. Also, JavaScript does not have colons at the end of the if-statement conditions.

**EXAMPLE:**

```javascript
if(age >= 21) {
  document.write("Ok to drink!");
}
```

JavaScript also utilizes **if/else** statements. It is essentially the same as an if statement but with a minor difference.

I. If the condition is **TRUE**, then the code inside the **first** set of curly braces is executed.
II. If the condition is **FALSE**, then the code inside the **second** set of curly braces (underneath the else statement) is executed.

**EXAMPLE:**

```javascript
if(age >= 21) {
  document.write("Ok to drink!");
} else {
  document.write("I'll stick with water.");
}
```
The \texttt{if/else} statement above can be illustrated using the following decision tree diagram:

![Decision Tree Diagram]

**Step 1:**
Let’s add \texttt{if} statements to our own code!

You might notice that if you have not had your birthday yet this year, your age calculation is wrong. We can easily fix that with the power of JavaScript!
   
   I. Create a new variable \texttt{birthday} 
   II. Prompt the user to ask them if they’ve had their birthday this year or not:

   ```javascript
   var birthday = prompt("Have you had your birthday yet?");
   if(birthday == "no") {
       age = age - 1;
   }
   ```

   **NOTE:** Recall the difference between equals signs. The ‘\texttt{==}’ (double equals) and ‘\texttt{=}’ (single equals) are the main two we will use. ‘\texttt{==}’ is used for COMPARING two values and ‘\texttt{=}’ is used for ASSIGNING values of variables.

**Step 2:**
Write an \texttt{if/else} statement to check if:
   
   I. Your age is greater than or equal to 18. If it is, output “I can open a checking account!” in a \texttt{document.write()}
   II. Your age is less than 18. If this is true, then output “I’ll just rely on my allowances.” in a \texttt{document.write()}
      
      Try it with different age values to make sure it works!
E. **Putting it all together**

We’re going to write a new program to ask users whether or not they have watched the Super Bowl. The webpage will output different questions/responses depending on how the person answers each question.

**Step 1:**
Using what you’ve already learned, follow along with your TA to create a program that

I. Creates a new “football” variable

II. Ask users if they have watched the super bowl this year

III. If the user has watched the super bowl, the program will ask them whether they are a Broncos fan. If didn’t watch the game, the program will ask if it was because they had a midterm coming up.

IV. Respond with some comment about their answer
Step 2:
Use an if/else structure to create the following program:

```javascript
var football = prompt("Did you watched the Super Bowl game this year?");

var broncosFan = prompt("Are you a Broncos fan?");
var midterm = prompt("Was it because you had a midterm coming up?");

if (football == "Yes") {
    if (broncosFan == "yes") {
        document.write("Must be a fun game to watch!");
    } else {
        document.write("Better luck next year!");
    }
} else {
    if (midterm == "yes") {
        document.write("Hope you Aced the midterm.");
    } else {
        document.write("Must feel good sleeping at a Sunday afternoon!");
    }
}
```

Step 3:
Start by using the structure above to complete the program. If you cannot do this on your own, you can find help below.

```javascript
var football;
var broncosFan;
var midterm;

football = prompt("Did you watched the Super Bowl game this year?");

if (football == "yes") {
    broncosFan = prompt("________");
    if (broncosFan == "yes") {
        document.write("________");
    } else {
        document.write("________");
    }
} else {
    midterm = prompt("________");
    if (midterm == "yes") {
        document.write("________");
    } else {
        document.write("________");
    }
}
```

Putting it online

Step 1:
Modify your CSE3Page.html to include a link to your SuperBowl.html page.

Step 2:
Put everything online and get checked off.

Checkoff: Go to your homepage via the class webpage and demonstrate to the TA/Tutor that your SuperBowl.html is complete. You must run through the program and show that ALL FOUR branches of your decision tree works.

REMEMBER to either drag your ENTIRE CSE3 folder into your public_html folder OR into the WHITESPACE inside the public_html folder!!!