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. GUI stands for graphical user interface that uses icons and other visual designs.

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 really interact dynamically with the user
   b. Can’t perform calculations

JavaScript is a solution to most of the HTML Cons 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 in that folder, named IntroJS.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. In the head tags, set the title to “JavaScript Trees” using the title tags. 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 the <center> tags in your <body> tags:

   <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 valid variable usage in JavaScript:

```javascript
var object;
var name = "Jackie";
var age = 21;
var isTrue = true;
```

Below are examples of invalid variable usage:

```javascript
var item // Invalid declaration - missing semicolon
width;
// Invalid initial declaration - missing "var"
length = 5;
// Invalid initial declaration - missing "var"
var height = 10;
// Valid int assignment of var "height"
var height = 11; // Invalid - cannot redeclare the same var
```

**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 the keyword `var`:

   ```javascript
   var name;
   var yearBorn;
   var age;
   ```

   b. Or you can combine them on one line and separate them with commas 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. Using these two new values, we can perform calculations to assign 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:
   1. `document.write();`
   2. `alert();`

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

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

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

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”
      
      **NOTE:** 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”
   d. If there are more than 2 items in the expression, follow the **order of operations**:
      i. `5.5 + 5.5 + "Super" = "11Super"`
      ii. `5.5 + (7.4 + "Super") = "5.57.4Super"`
      iii. “Super” + 5.5 + 7.4 = “Super5.57.4”

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

To get information from the user, we can use the `prompt` function to ask for the user’s input.
**Step 1:**
Modify your program to ask for the user’s name and yearBorn using the prompt function:

```javascript
var name = prompt("What is your name?");
var yearBorn = prompt("What year were you born in?");
```

Try it and see what happens!

**Step 2:**
Use concatenation to create a sentence with the name and age; display it with 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 that gets printed.

II. name and age are considered to be variables, so they must go OUTSIDE the quotation marks. This way, JavaScript knows to look for their assigned value instead of just printing the texts “name” and “age”.

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

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. **NOTE:** Unlike Python, JavaScript utilizes curly braces and does not use colons.

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 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.
Step 1:
Let’s add 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 fix that with JavaScript!

I. Create a new variable called 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 ‘==’ (double equals) and ‘=’ (single equals) are the main two we will use. ‘==’ is used for COMPARING two values and ‘=’ is used for ASSIGNING values on the right hand side to the variables on the left hand side.

Step 2:
Write an if/else statement to check if:

I. Your age is greater than or equal to 21. If this is true, output “OK to drink!”
II. Your age is less than 21. If this is true, output “I’ll stick with water.”
III. Try it with different age values to make sure it works!

```javascript
if(age >= 21) {
    document.write("Ok to drink!");
} else {
    document.write("I'll stick with water.");
}
```

The if/else statement above can be illustrated using the following decision tree diagram:

![Decision Tree Diagram](image-url)
Task: We’re going to write a new program to ask users whether they are graduating this quarter and what their plans are for after graduation. It will output different questions/responses depending on how the person answers each question.

Step 1:
Create a new file called Graduation.html in Notepad++ and save it in your Lab7 folder.

Step 2:
Using what you’ve already learned, follow along with your TA to create a program that:

I. Creates a new “graduation” variable.
II. Ask users if they are graduating this quarter.
III. If the user is graduating, the program will ask them if they are planning on going to grad school.
IV. If the user is not graduating, the program will ask them what year they will be in next year: 2\textsuperscript{nd}, 3\textsuperscript{rd}, or 4\textsuperscript{th} (and beyond).
V. Respond with some comment about their answer.
**Step 2:**
Use statements of the `if/else if/else` structure to create the following program:

```javascript
var graduating = prompt("Are you graduating this quarter?");

var grad_school = prompt("Are you planning on going to grad school?");

var year = prompt("What year are you going into?");

if (graduating == "yes") {
  if (grad_school == "yes") {
    document.write("All your hard work will pay off!");
  } else {
    document.write("I hope you find a job that you love!");
  }
} else {
  if (year == "2") {
    document.write("You still have a lot of time left, enjoy yourself!");
  } else if (year == "3", "4", "5", ...) {
    document.write("You're halfway there!");
  } else {
    document.write("Almost done!");
  }
}
```

**Step 3:**
Use the question structure above and the JavaScript template below to complete the program.
**NOTE:** Conditional statements and strings are case-sensitive so be careful with your answers.
Checkoff:
Go to your homepage via the class webpage and demonstrate to the TA/Tutor that your Graduation.html is complete. You must run through the program and show that **ALL FIVE** branches of your decision tree work.

**Putting it online**

**Step 1:**
Modify your **CSE3Page.html** to include a link to your **Graduation.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 Graduation.html is complete. You must run through the program and show that **ALL FIVE** branches of your decision tree work.

**REMEMBER** to drag your CSE3 folder into the WHITESPACE inside your **public_html** folder!