Kelly Kuhn

N220

July 6th, 2023

**Lab 7 – User Input Exercises**

<https://rkkuhn.github.io/N220Spring2023/>

**Algorithm 1st Project – Tip Calculator**

**Requirements:**  Make a page with one input and one button (Seeing a trend?). When the button is clicked (assuming the value typed into the input is numerical), calculate both the tip and the full value of the bill to be paid. Output both to the console, formatted as such:

"Tip: $X. Total: $X"

**Expected Output:**

A screenshot of a computer

Description automatically generated

**Sudo Code:**

HTML

Create the heading for the tip calculator

Ask user to enter bill amount

Input save from users response

Create button to calculate the bill tip and total amount to be paid

JS

Get the value entered for the bill amount

Check if the input is a valid number if not ask the user to try again

Calculate the tip at 20% or 0.20

Calculate the total bill including the tip

Output of the tip amount and total to be paid goes to the console per the directions above

**Algorithm 2nd Project – Number Guessing Easier**

**Requirements:** Create a number guessing name, using an input and a button to gather a number.  
The number to be guessed should be a hard-coded whole number between 1 and 20.  
Tell the user if the number is too high, equal to, or too low than a number you have hard-coded in your application.

Remove the text in the input when the user clicks the button.

**Expected Output:**

A screenshot of a game

Description automatically generated

A screenshot of a game

Description automatically generated

**Sudo Code:**

HTML

Create the header

Text for user to read about the game. “Guess a number between 1 and 20”

The input from the user w/ a place holder inside the entry box

Onclick button submission

Id=”result”

JS

Hard-coded number to be guessed between 1 and 20

Create function checkGuess using document.getElementById

Set up the if statement, else if statements and else statement.

Reset the input field back to the place holder

**Algorithm 3rd Project – Greeter**

**Requirements:** Create a page with an input and a button. When the button is clicked, output the phrase “Hello {Name}” to the developer console, with {Name} being the value the user put into the input field.

**Expected Output:**

A white background with black text

Description automatically generated

**Sudo Code:**

HTML

Create the required fields for the button

Ask the user to input their name and submit using oneclick

<label for = “nameInput”>Enter your name:</label>

<input type=”text” id=”nameInput”>

<button onclick=”sayHello()”.Submit</button>

JS

This was the easier part of the code

Create a function to get the input value “nameInput” using document.getElementByID

Submit the out to the console.log per the instructions (“Hello “ + name);

**Algorithm 4th Project – Password Protected**

**Requirements:**  Create an application with two input fields and one button.

When the button is clicked, verify the user has written these (exact) strings in the first and second field:

* First field should be  "Username"
* Second field should be  "Password"

If the two fields match, update a DIV on the page with the text "Success" or else "Wrong information".

For this assignment: use only one if statement to to the check.

Remember you'll need to use the double-equals (==) to check for equality, and the double-ampersand to check if two things are true.

if( (oneThingIsTrue) && (anotherThingIsTrue) ) {

}

**Expected Output:**

A screenshot of a login box

Description automatically generated

**Sudo Code:**

I will not put the sudo code here for two reasons: 1) This is a bonus from my twisted mind. 2) I want it to be a surprise when you get the username and password wrong.

The correct username and password are Username and Password.

I hope you enjoy my warped sense of humor.

***Reflection:*** When doing these projects, my head always wants to go to the first place I know, HTML. I know there are always multiple ways to program something, yet at first, when I approach JavaScript projects, I always head down the HTML path. I must go back over my notes after I read the instructions to ensure I am writing it in a different format than HTML and then pull out what should go to JS. I get the same results either way, but it wastes time doing that. I should only write code in two languages when I am genuinely stuck and know I can do it in another language as a jumping-off point. I have used my HTML skills as a crutch for far too long and need to start taking steps toward the JS light first.