Name _____ Period _____ Role (Circle one) Programmer/Driver

Name _____ Period _____ Role (Circle one) Programmer/Driver

Variables and user input

Your Tasks (Mark these off as you go)

- ☐ Create and initialize variables
- ☐ Have Ms. Pluska check off the above tasks
- ☐ Apply arithmetic operations
- ☐ Apply concatenation to join variables
- ☐ Prompt the user for input
- ☐ Have Ms. Pluska check off the above tasks
- ☐ Receive credit for the individual portion of this lab

☐ Create and initialize variables

Variables are data types that we use in programming. Variables are essential for controlling the memory in our programs. Watch the video below to learn more about variables (https://www.youtube.com/watch?v=G41G_PEWFjE)

Indicate whether each of the following variable declarations are legal or illegal	
code	**legal/illegal**
var 1231abc;	
var big-number;	
var name2;	
var myVar*2 = 0;	
2 = var myNum;	

Write code to, (a) To declare a variable called score (b) To initialize the variable above to 0 (c) Log the “The value of score is : 0” to the console, where 0 is the variable score.

It's so common to want to create a variable and give it an initial value, that JavaScript has a shortcut that lets you create and assign with one line of code like this:

```
var score = 0;
```

Write code to decare and intialize a variable called lives. Assign lives to zero.

The above examples illustrate how to store numeric data in memory. In javascript, we can also store String type variables. A String is any grouping of characters on your keyboard (letters, numbers, spaces, symbols, etc.) surrounded by single quotes: ' ... ' or double quotes " ... ". Though we prefer single quotes. Some people like to think of string as a fancy word for text. Consider the following examples

```
var name;
name = “wigglesworth”;
console.log("My name is ");
console.log(name);
```

Write code to, (a) Declare a variable called name and assign your name to it. (b) Create a new variable called about, then assign a sentence about you to this variable. Do this on one line. (c) Log your name and about variables to the console.

☐ Have Ms. Pluska check off the above tasks

Before you continue have Ms. Pluska check off the above tasks

Do not continue until you have Ms. Pluska’s (or her designated TA’s) signature _____

☐ Apply arithmetic operations

Basic arithmetic often comes in handy when programming.
An operator is a character that performs a task in our code. JavaScript has several built-in in arithmetic operators, that allow us to perform mathematical calculations on numbers. These include the following operators and their corresponding symbols:

- Add: +
- Subtract: -
- Multiply: *
- Divide: /

- Remainder: %

The first four work how you might guess:

```
console.log(3 + 4); // Prints 7
console.log(5 - 1); // Prints 4
console.log(4 * 2); // Prints 8
console.log(9 / 3); // Prints 3
```

Note that when we console.log() the computer will evaluate the expression inside the parentheses and print that result to the console. If we wanted to print the characters 3 + 4, we would wrap them in quotes and print them as a string.

The remainder operator, sometimes called modulo, returns the number that remains after the right-hand number divides into the left-hand number as many times as it evenly can: 11 % 3 equals 2 because 3 fits into 11 three times, leaving 2 as the remainder.

Write code to, (a) Create a new variable called age and assign your age to this variable (b) Inside of a console.log(), add 3 to your age. This is the age you'll be when we start sending people to live on Mars. (c) On a new line write another console.log(). Inside the parentheses, take the current year and subtract 1969. The answer is how many years it's been since the 1969 moon landing. (d) On a new line write another console.log(). Inside the parentheses, multiply 0.2708 by 100. That's the percent of the sun that is made up of helium. Assuming we could stand on the sun, we'd all sound like chipmunks! (e) Create on last console log, print the number that remains when your age is divided by 10.

□ Apply concatenation to join variables

Operators aren't just for numbers! When a + operator is used on two strings, it appends the right string to the left string:

```
console.log('hi' + 'ya'); // Prints 'hiya'
console.log('wo' + 'ah'); // Prints 'woah'
console.log('I love to ' + 'code.')
// Prints 'I love to code.'
```

This process of appending one string to another is called concatenation. Notice in the third example we had to make sure to include a space at the end of the first string. The computer will join the strings exactly, so we needed to make sure to include the space we wanted between the two strings.

```
console.log('front ' + 'space');
// Prints 'front space'
console.log('back' + ' space');
// Prints 'back space'
console.log('no' + 'space');
// Prints 'nospace'
console.log('middle' + ' ' + 'space');
// Prints 'middle space'
```

Just like with regular math, we can combine, or chain, our operations to get a final result:

```
console.log('One' + ', ' + 'two' + ', ' + 'three!');
// Prints 'One, two, three!'
```

Write code to (a) Concatenate two strings 'Hello' and 'World' inside a console.log statement (b) console.log() 'Hello' and 'World', but make sure to use string concatenation to also include a space (' ') between the two words.

☐ Prompt the user for input

Programs become even more interesting when we can interact with the user. A short way to ask a user for information is with the prompt command, which pops up a dialog box asking the user for input. Consider the example below. The example below prompts the user for their first and last name, then prints a message to the console.

```
var firstName = prompt('Enter your first name');
var lastName = prompt('Enter your last name');
var fullName = firstName + ' ' + lastName;
console.log('Hello' + fullName);
```

Use the example above to write code to (a) Prompt the user for additional information. For example, “What is your favorite food?”, “What is the most interesting place you have visited?”. (b) Write code to print the user’s name, along with the additional information to the console. Make sure it is properly spaced and legible.

☐ Receive Credit for the group portion of this lab

- ☐ Make sure both you and your partner have completed the above tasks
- ☐ Indicate the names of all group members.
- ☐ Have Ms. Pluska check off group tasks
- ☐ Submit your lab to the needs to be graded folder to receive credit for the group portion of this lab.
- ☐ Do not submit your lab until you have Ms. Pluska’s (or her designated TA’s) signature

_____