**DEBUGGING**

**Commenting:**

*//Line by line commenting  
//even on the next line*

*/\* commenting a multi-line text or paragraph  
continued text…  
continued text….  
more continued text…VARIALBLES \*/*

/\*The **console** is available in your browser where you can run code directly. It also allows you to see the output of JS you write in your editor. *(***Quick key***: A combination of CTRL, Option, Shift “J)*

**Alerts** are pop-up windows. *Alert()* function gets called.

**Logging to HTML** means to change the content of an HTML element with content of your choice, which could be script output.

**Why check?** To see what your *code return,* is actually want you want it to return.\*/

**//Alert Function**//Alert {“Hello World!”);

**/\*Logging Directly to HTML**

function change(){

document.getElementById(‘el’).innerHTML = “NEW TEXT”;

}

<div onclick=”change()“ id=”el”>CHANGE ME</div>

// Line 1: Declare a function called change

**function change(){**

// Line 2: Inside of the function, retrieve the current

// HTML document. Inside of that document, retrieve

// the element with the ID ‘el’ using the getElementById

// function. Set the content inside of the element to say

// “New Text”.

**document.getElementById(‘el’).innerHTML = “NEW TEXT”;**

// Line 3: Close the function.

**}**

// In your HTML document, you’ll need a <div> with the ID

// “el”. The onclick attribute allows you to specify a

// function to run when the element is clicked, in this

// case “change()”

**<div onclick=”change()“ id=”el”>CHANGE ME</div>\*/**

**DATA TYPES**

**/\*STRING**  
by encapsulating our text into “quotes”, we are telling JavaScript the data inside is **String,**  
 A basic JavaScript data type. \*/

**/\*Data Types:**

* String - “Hello World!”
* Number - 5, 5.5, 1000 (All numbers in JS are floats)
* Boolean – true, false
* Undefined (no value)\*/

**//Variables**

//A variable in simply a container for a value

X = 0

X + 10 = ?

Answer: 10

//Variables are defined as such:

Var name = “Zach”;

Var numberOfWidgets = 10;

Var isCodingCool – true;

**//Basic Math**//Code below to be run in Console:

10 + 10;

20

var x = 100;

x \* 40;

4000

//Implementing commenting below

//**Further Data Types**

//Arrays are used to hold a collection of data, of any data type. Below is full of strings:

[“Ford”, “Chevy”, “BMW”, “Mercedes”];

//**They can hold multiple data types:**

[11, 15, “BMW”];

//**Arrays can also be stored in variables:**

Var myFamily [“Christine”, “Jensyn”, “Christopher”, “Jonathan”, “Zachary”];

//END OF COMMENTING

**Accessing Array Items - Indexes**

* Once you have declared an array, you may want to retrieve the items inside of it using their indices
* The index of an item inside of an array corresponds to its position from the beginning of the array
* The first item always has an index of 0
* In below array “John Oertel” has the index of 0 and “Christine Oertel” has the index of 1:  
  [“John Oertel”, “Christine Oertel”]

**An Array can store other arrays**

var fords = [“fiesta”, “focus”];

var chevys = [“camaro”, “vette”]

var cars = [fords, chevys];

The above is a multi-dimensional array

**Declare your multi-dimensional array**

var cars = [ [“fiesta”, “focus”], [“camaro”, “vette”] ];

**Access the first array inside, then the first item inside that array**

Console.log(cars[0][0]);

*>fiesta*

**TESTING**

Any test returns a Boolean *true* or *false*

*To test if two strings are equal: “stringone” == “string two”;*

*>false*

Using 3 equal signs instead of 2 also checks the object type ===

If you don’t check type, these are both true:

*(10-5) == 5;*

*(10-5)== “5”;*

*The above can cause bugs down the road!*

To test if two strings are not equal:

*“stringone” !== “string two”;*

*>true*

To test if 1 number is greater than another:

*5> 10;*

*>false*

Note: Other operators are also valid comparisons

*<, >, <=, >=*

**If…Then…Else…Then**

*if(5>10){*

*console.log(“You’ll never see this*

*because 5 is not greater than 10”);*

*} else{*

*console.log(“You will see this because*

*5 is not greater than 10”);*

*}*

**If...Then...Else If...Then...Else...Then**

Else if is another condition to evaluate in the case where if is not true and you have another condition to look at before else:

*if(5>10){*

*console.log(“You’ll never see this because 5 is not*

*greater than 10”);*

*}else if(5===5){*

*console.log(“Yes, 5 really is equal to 5.”)*

*}else{*

*console.log(“You will see this because 5 is not*

*greater than 10”);*

*}*

**FUNCTIONS**

A function is a way to encapsulate code for later use

It can take **arguments**, which are used as variables inside the function

*function addTwo(some\_number){*

*return some\_number + 2;*

*}*

*console.log(addTwo(4));*

*>*

Once a function is declared, it can be called later on (invoked) by calling its name and   
supplying it with any arguments

*function alertName(somePersonsName){*

*return alert(somePersonsName);*

*}*

*alertName(“Zach”);*

**IDENTIFYING DATA TYPES**

Assuming you have a variable with some data stored inside of it, and are unaware of its data type…

You can ask (or query) the variable for its datatype using the typeof() function

*var yourData = “This is my data.”;*

*typeof(yourData);*

*>string*

**Accessing Array Items**

If you’re unsure of an items index number inside of an array, When unsure of an index number

*var snoopyPosition = myArr.indexOf(“Snoopy”);*

*console.log(myArr[snoopyPosition]);*

*>”Snoopy”*

**LOOPS**

A loop is a block of code that gets repeated for a defined amount of iterations (for loop)

or until a certain condition is met (while loop) Typically one variable or condition in the loop changes each time it is run

**Loops - while**

x = 6

while(x < 10){

console.log(“On number “ + x)

x++;

}

>>6

789

**Loops - for**

*for(var i = 0; i<10; i++){*

*console.log(i)*

*}*

*>>0*

*123*

*...*

*9*

beers = [“Lagunitas”, “Peak”]

for(var i = 0; i< beers.length; i++){

console.log(beers[i])

}

>>”Lagunitas”

“Peak”