**JavaScript**

To enter JavaScript you can open a browser on a page (usually incognito) and right click > inspect element. Then go to console tab and you can start entering in JavaScript.

**Built-in Functions**

clear() - clears console

alert(x) - Pops up an alert with the parameter as a message

prompt() - Takes in input from user

**Basic Data Types**

Numbers

Strings

Booleans

Undefined

Null

**Variables**

Python:

variable\_name = value

JavaScript:

var varName = value;

**Re-assigning a Variable**

If a variable has already been assigned and you are re-assigning the value inside of a loop or function you can use the python method of variable assignment.

Ex:

varName = newValue;

NOTE: Naming format for variables is camel casing.

**Useful Functions**

**Print to Console**

Python:

print(“hello console”)

JavaScript:

console.log(“hello console”);

**Taking in User Input**

Python:

input(“Input here”)

JavaScript:

prompt(“Input here”);

**Getting Item Length**

Python:

len(“string”)

JavaScript:

“string”.length;

**Operators**

**Comparison Operators**

**Equal to**

Python:

x **==** y

JavaScript:

x **===** y;

**Not equal to**

Python:

x **!=** y

JavaScript:

x **!==** y;

**Logical Operators**

**AND**

Python:

x > y **and** x < z

JavaScript:

x > y **&&** x < z;

**OR**

Python:

x > y **or** x < z

JavaScript:

x > y **||** x < z;

**NOT**

Python:

**not** x

JavaScript:

**!**(x);

**JavaScript Strings and Numbers Comparisons**

In JavaScript when comparing the string value of a number and a integer value of a number with “==” the result will be true.

Ex:

2 == “2”;

Result is true

2 != “2”;

Result is false

Example using the correct method:

2 === “2”;

Result is false

2 !== “2”;

Result is true

NOTE: This is the same for all operators.

**Control Flow**

**IF**

Python:

if condition:

execute code

JavaScript:

if (condition) {

execute code;

}

**IF ELSE**

Python:

if condition:

execute code

else:

execute other code

JavaScript:

if (condition) {

execute code;

}else{

execute other code;

}

**IF, ELSE IF, ELSE**

Python:

if condition:

execute code

elif condition2:

execute condition2 code

else:

execute other code

JavaScript:

if (condition) {

execute code;

}else if (condition2) {

execute condition2 code;

}else{

execute other code;

}

**Loops**

**While**

Python:

while condition:

execute code while condition is true

JavaScript:

while (condition) {

execute code while condition is true;

}

**Using a While Loop with IF Statement**

var x = 0;

while (x < 5) {

console.log(x);

if (x === 3) {

break;

}

x = x + 1;

}

**For Loops in JavaScript**

JavaScript has 3 types of For Loops:

- for - loops through a number of times

- for/in - loops through a JS object

- for/of - used with arrays

**For**

Python:

for x in range(y):

code

JavaScript:

for (statement1; statement2; statement 3) {

code;

}

**Breakdown of Statements**

Statement1 is executed before the loop starts.

Statement2 defines the condition for running the loop.

Statement3 is executed each time after the loop cycles through.

Ex:

for (var i = 0; i < 5; i++) {

code;

}

In the example above the for loop will will start at 0 (var i = 0;) and loop 5 times (i < 5) with an increment of 1 (i++).

**Using a For Loop to Loop Through a String**

var word = "ABCDEFGHIJK";

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

console.log(word[i]);

}

**For/of**

Used to iterate through arrays.

Python:

for x in list:

code

JavaScript:

for (x of array) {

code;

}

**Using a Function With Each Item in An Array**

array.forEach(function);

This is a shorthand for:

for (x of array) {

function(x);

}

**For/in**

Used to iterate through JavaScript Objects.

Python:

for key in my\_dictionary:

code

Or

for key, value in my\_dictionary.items():

code

JavaScript:

for (key in myObject) {

code;

}

**Functions**

Python:

def func\_name(parameters):

code

JavaScript:

Function funcName(parameter1, parameter1) {

code;

}

NOTE: You can use “return” like you would in Python.

**Arrays**

Arrays are the equivalent of a list in Python.

**Creating An Array**

var name = [item1, item2, item3];

**Modifying An Item in An Array**

var name = [item1, item2, item3];

name[2] = newItem;

Result of name will be [item1, item2, newItem]

**Adding to An Array**

Python:

My\_list.append(item)

JavaScript:

myArray.push(item);

**Using the Pop Method**

Same for both Python and JavaScript.

JavaScript:

myArray.pop();

**Retrieving the Last Item From An Array**

Python:

my\_list[-1]

JavaScript:

myArray[myArray.length - 1]

**Finding the Index of An Item**

Python:

my\_list.index(list\_item)

JavaScript:

myArray.indexOf(arrayItem);

**Deleting an Item From An Array**

Python:

my\_list.remove(list\_item)

Or

del my\_list[index]

JavaScript:

You can remove an item from an array by its index (useful in conjunction with .indexOf method).

myArray.splice(index, 1);

**Check to See if Item is in Array**

Python:

list\_item in my\_list

JavaScript:

myArray.includes(arrayItem);

**JavaScript Objects**

JavaScript Objects are hash-tables that store information in a key-value pair. The equivalent of a dictionary in Python.

**Creating a JavaScript Object**

var myObject = {key1: value1, key2: value2, key3: value3};

NOTE: Quotation marks are NOT used on the key names when creating an object. (Values still use quotation marks when using strings.)

**Retrieving an Object Value**

myObject[“key1”];

NOTE: Quotation marks ARE used when retrieving the object value.

**Changing an Object Value**

myObject[“key”] = newValue;

**JavaScript Object Methods**

JavaScript object methods are functions inside an object.

**Creating An JS Object Method**

var myObject = {key: function(){myFunction(arguments)}}

Think of the key as the method name.

**Example of an JS Object Method:**

var myObject = {

prop: 37,

reportProp: function() {

return this.prop;

}

}

Enter in console:

console.log(myObject.reportProp());

Result will be 37

Another Example:

var userObject = {

firstName: "John",

lastName: "Smith",

fullName: function() {

console.log("The users full name is " + this.firstName + " " + this.lastName + ".")

}

}

Enter in console:

userObject.fullName()

Result will be “The users full name is John Smith.”

You will need to use “**this**” keyword when referencing a key inside of the object method. (“This” appears to be complicated so read into more.)

NOTE: Functions can be placed as values in a dictionary in Python as well.

**Other Notes**

1. There are 3 separate ways to add 1 to a variable:
   1. var = var + 1;
   2. var += 1;
   3. var++;
2. JS treats all numbers (integers and floats) as the same.
3. To comment code use two forward slashes at beginning of comment. Ex: // Here is a comment.
4. To connect a “.js” file to an “.HTML” file insert the following:
   1. <script src=”yourfile.js”></script>

**Resources (Have to right-click and open hyperlink)**

[Mozilla Web Docs](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide)

[W3Schools](https://www.w3schools.com/js/default.asp)