Template literals notion using the back tic ` to open up the string $ { allows you to put in variables etc.}

**Javascript**

Note: with javascript. The first later is always lowercase.  
Data Types

| **Variable** | **Explanation** | **Example** |
| --- | --- | --- |
| [String](https://developer.mozilla.org/en-US/docs/Glossary/String) | A sequence of text known as a string. To signify that the value is a string, you must enclose it in quote marks. | let myVariable = 'Bob'; |
| [Number](https://developer.mozilla.org/en-US/docs/Glossary/Number) | A number. Numbers don't have quotes around them. | let myVariable = 10; but if you but in quotes it can be a string. |
| [Boolean](https://developer.mozilla.org/en-US/docs/Glossary/Boolean) | A True/False value. The words true and false are special keywords in JS, and don't need quotes. | let myVariable = true; if but into quotes it is again a string |
| [Array](https://developer.mozilla.org/en-US/docs/Glossary/Array) | A structure that allows you to store multiple values in one single reference.  1st item in an array is always 0. | let myVariable = [1,'Bob','Steve',10]; Refer to each member of the array like this: Arrays are always [ ] myVariable[0], myVariable[1], etc. |
| [Object](https://developer.mozilla.org/en-US/docs/Glossary/Object) | Basically, anything. Everything in JavaScript is an object, and can be stored in a variable. Keep this in mind as you learn. | let myVariable = document.querySelector('h1'); All of the above examples too. |

Var age = 10 Var=Variable then you need to declare the variable in this case “10” so in the program anytime you use the word age it =10

Consule.log(age) if you then write the Console.log(age + age) it will = 20

You have two main data types

1. Primitive
   1. Strings var name=’joe’
   2. Numbers var age=20 Don’t use quotes, if you do it is a string
   3. Booleans = true or false
      1. Var isAlive = true (if you put in quotes this make it a string)
      2. hasDoneTaxes = false
2. Complex= a collection of data types
   1. Arrays = [1,2,3,4] List of similar items
      1. Var numbers= [ 1, 2, 3 ] need to be comma separated
      2. Numbers [0] indexing starts at zero
   2. Objects Explains a singular item in depth (key value pairs) 1st you need to name the
   3. Var person = {
      * 1. Name: “Steve”, Note: **: : must have a colon, it is a STRING must have ‘ quotes ‘ and must have a ,**
        2. Age: 21, Number **: must have a colon, comma ‘,’ no quotes**
        3. hairColor: “brown”, **: must have a colon, ‘,’ quotes and a comma it is a STRING**
        4. isAlive: true NOTE **for very last item there is no comma. This is a Boolean so it must be true or false**

**Example** Var ourStr = “I come first. “ + “I come second.”;

Var myStr = “This is the start. “ + “This is the end.”

Console.log(myStr);

**Example** var myStr = “This is the first sentence. “

myStr=+”This is the second sentence.“

console.log(myStr);

**Arrays**

Var myNumbers = [1, 2, 3, 4, 5]

0 1 2 3 4

Var colors = [“blue”, “green”, “red”, “yellow”, “purple”]

Console.log(colors[1])

Every Array has a length

Console.log( colors.length) answer is 5

Strings also have length property  
var name = “joe”

Console.log(name.length)

name[0]

**Conditionals**

If(true) {

Console.log(It is true!”)

}

If(false) {

Console.log(It is true!”)

} else {

Console.log (It is false!”)

If (2 === 2) {  
console.log(“Two is equal to two”)

} else {

Console.log(Two is not equal to two”)

}

If (2 === 4) {  
console.log(“Two is equal to two”)

} else {

Console.log(Two is not equal to four”)

}

If (2 === 4) {  
console.log(“Two is equal to two”)

} else {(2 === 5){

Console.log(Two is not equal to five”)

} else {(2 === 2){

Console.log(Two is not equal to two”)

}

If (2 === 4) {  
console.log(“Two is equal to two”)

} else {(2 === 5){

Console.log(Two is not equal to five”)

} else {(2 === 6){

Console.log(Two is not equal to six”)

} else {  
console.log(“I don’t know what is going on”)

}

If (“2” == 2){  
console.log(“They are equal”)

} This would be true REMEMBER THAT “2” is a STRING so loosely it is equal to 2 as 2 matches 2. If it was === strict then it would be False.

There are Truthy and Falsey

All Falsey

* 0
* “ ”
* Null
* Undefined
* False
* NaN not a number

If(“”) {

console.log(“it is truthy”)

} else {

console.log(“it is falsey”)

}

If you were to put a character into “a” then it would be truthy

**Logic Operators**

&& - and

|| - or

! – not

If (2 === 2 && 2 === 3) { **NOTE: This both needs to be true for the && operator to work**

console.log(It’s working!”)

} else {

console.log(“Its’s not working!”)

}

If (2 === 2 || 2 === 3) { **NOTE: This both needs to be true for the && operator to work**

console.log(It’s working!”)

} else {

console.log(“Its’s not working!”)

}

If (2 === 2 && 2 !== 3) {

console.log(It’s working!”)

} else {

console.log(“Its’s not working!”)

}

**Order of Operations**

+ - \* /

Example If (2 === 2 && 3 === 3) && 2 !== 3){

console.log(It’s working!”)

} else {

console.log(“Its’s not working!”)

}

**Switch**

Here is a Switch Statement

Var color = “blue”

Switch(color) {  
 Case “red”:  
console.log(“The color is red”)  
**}** take out then add break this will stop the code each time

Case “blue”:  
console.log(“The color is blue”)  
break

Case “yellow”:  
console.log(“The color is yellow”)  
break

Default:

console.log(“The color is not red, blue, or yellow”)

}

var person = “Bobby”;  
var age= “12”   
}

if (“age”<=18) {  
console.log( person + “ is old enough to go to this movie!”)  
} else {  
Console.log( person + “ is not old enough to go to this movie!”)  
}age

var person = “Bobby”;  
var age= “12”

if (“age” <= 18) {

console.log(“ + person + “is old enough to go to this movie!”)

} else ("age" > 17) {

Console.log(“ + person + “is not old enough to go to this movie!”)  
}

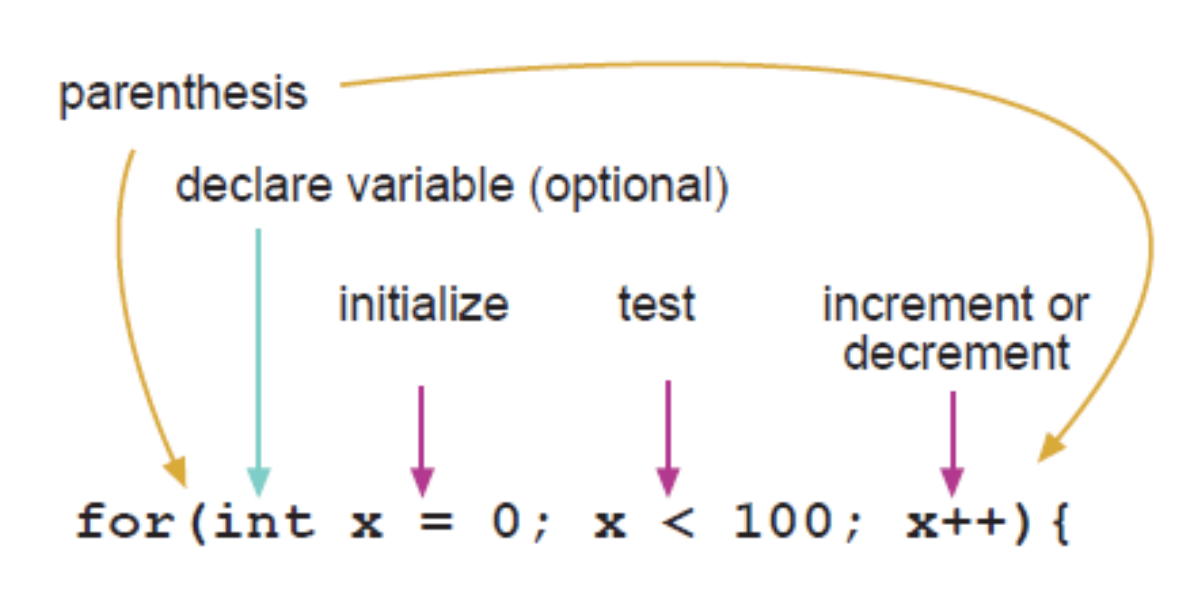
If (2 === 4) {  
console.log(“Two is equal to two”)

} else {

Console.log(Two is not equal to four”)

}

**Java Script Loops**

****

**You must make the for loop stop otherwise it will continue.**

**NOTE: DO NOT USE CAPITAL LETTERS FOR THE START OF ANYWORD IT WILL BREAK THE CODE AND GIVE YOU ERRORS THAT AREN’T REALLY ERRORS.** Example below For is Capitalized and it will give you 3 error messages.

**Breakdown of this code and how it works For(let i = 0; i < 100; i ++){**

**Console.log(i)**

**}**

1. Declare variable but put let in instead of var.. **1st part of statement** **for(let i = 0;**
   1. What number do you want to start counting at? Start at 0
   2. ‘ i ‘ states for iterator, it is the thing that is going to be counted 1 by 1, 2 by 2, etc.
   3. NOTE: each of these codes are set up by a ‘ ; ‘
2. Tell it where to stop. **2nd part of statement i < 100;**
   1. ‘ i ‘ states for iterator, it is the thing that is going to be counted
   2. < 100 “ less than 100 ” determines where the “ i “ is going to stop. It will run until it reaches 99 than stops.
   3. NOTE: each of these codes are set up by a ‘ ; ‘
3. Last item describes how you want me to count by
   1. Using the i++ equation. This equation increase to +99
   2. Using the i—equation. This decreases to -99
   3. How many times will did this code run “ 100 “ because it counted to 100

In short.

1. What number do I want to start counting at? **STARTING POINT**
2. Where do I want it to end? **STOPPING POINT**
3. Every time it counts how many times to, I want it to increment by; or decrement by;
4. for ([initialExpression]; [condition]; [incrementExpression])
5. statement

**Example of Loop**

for(let i = 0; i < 100; i ++){

console.log(i)

}

**Example of an Infinite Loop**for(let i = 25; i === 20; i ++){

console.log(i)  
}

This will start counting at 25, go to 26, 27 etc. and never stop because it will not reach 20. It will crash

Var favfoods = [“pizza”’ “pasta”’ “ice cream”, “banana”]

favFoods.length //4

Typing in the code this way is way to repetitive and a waste of time.

\Console.log(favFoods[o]) Give me pizza

\Console.log(favFoods[1]) Give me pasta

\Console.log(favFoods[2]) Give me ice cream

\Console.log(favFoods[3]) Give me banana

For(var i =0; I < favfoods.length; i++) {

Console.log( favFoods[i] ) **This is really important syntax**

}

Var numbers = [1, 2, 3, 4, 5, 6]  
numbers[0]

For(var i =0; I < numbers.length; i++) {

If (numbers[i] % 2 === 0) [

Console.log(numbers[i] )

}

Var count = 0

While(count <10) {  
console.log(‘hi’)

Count++

}

**Functions** DRY = Do Not Repeat Yourself

Statement is any line of code

Var name= “nate”

**(This is the function Declaration)**

Function sum() {

Console.log(2+2)

}

**(Function Expression)**

var mySumFunction = function () {

Console.log (2+2)

**(Parameters)**

Function sum(num1, num2) {

Return num1 + num2

}

//We are calling the function or execute the function

Var result =sum(10, 25) arguments

Console.log(result)

Function myFunc(data) {

Console.log(data)

}

Myfunc(1)

Myfunc(“hello”)

Myfunc(true)

Myfunc([1, 2, 3, 4])

Myfunc({name: “joe”})

Function loopThroughArr(array){

For (var i = 0; i < array.length; i++){

Console.log(array[i]

}

Objects best used to describe something in-depth

Key: value

Var person = {

Name: “Rick”

Age: 70

Friends: [ “Morty”, “Joe”, “Sam”, “Samantha”],

Address: {“123 street”,

City: “Somewhere”

}

Two notations for accessing an object’s data

* Dot “.”

Console.log(person.name)

Console.log(person.age)

Console.log(person.friends)

Console.log(person.address.street)

* Bracket notation [ ]

Console.log( person [“address”] [“city”]

Var car = {

Type: “Honda”  
make: “Civic”  
wheels: 4  
  
Honk: function(){  
Console.log( car [“Honk”]  
}

Car.honk()

//Car.hasHadAccident=true  
// Console.log( person (car)

**Objects can hold methods**

Var car = {

Type: “Honda”  
make: “Civic”  
wheels: 4  
honkSound: “Blleerrp”’  
Honk: function(){  
Console.log( this.honkSound ) As “THIS” represents “CAR”  
}

How are objects referenced

Objects and Arrays are passed by reference

Var othercar = car

**Conventions**

1. Casing= camelCased 1st word is lowercase secondword is uppercase
   1. Var lastName
   2. Var myLocationIsInUtah
2. White space
   1. For(var i = 0; i < 10; i++){ use an extra space so you can read the code more readily.
3. Naming
   1. Var firstName = “Nate”
   2. Var lastName = “Jensen”
   3. Array ‘s can be different
      1. var friends = [ ‘ john ‘, ‘ eric ’ ]
      2. var isDone = true
      3. var isGameOver = false
      4. function sum(num1, num2){

}

1. Braces
   1. Hit enter to keep everything lined up the Visual Studio will intend everything for you.
      1. Var person ={ enter
2. Tab (Indenting)
   1. If (2 ===2) {

Console.log(“they are equal”)

If(3 ===3) {

Console.log(“they are equal”)

}  
}

1. HTML Tags
   1. Use lower case <div>
2. Comments
   1. //
3. Unix casing
   1. Lower case and naming conventions, silly\_cia

**What is a Dom? Document Object Model**

**Javascript object that represents the document or webpage, it provides a set of tools for the developer to use to manipulate the web page.**

**Index.js**

**document**

**Event Listener**

**The Standard Loop for running conditional statement with math**

**For (initializer; exit condition; final-expression0 {**

**// code to run**

**}**

**Counter, which is initialized with a certain value- this is the starting point of the loop. (“Start: I have no food, above).**

**Exit condition, which is the under which the loop stops- usually the counter reaching a certain value.**

**An iterator, which is generally increments the counter by a small amount on each successive loop**

**Fr**

**In the for statement**

**Var is what initiates the condition**

1. **String: “ “,**
2. **Number 10,**
3. **Boolean**

**Complex**

1. **Array:** [1,2,3,4] List of similar items
   1. String Array is [“Bob”, “Rick”, “Jack”, “Tom”]
   2. Numbered Array [ 1, 2, 3 ] need to be comma separated
      1. To start an array you must note that it always starts [0] indexing starts at zero, so if you have 1, 2, 3, 4 then there are how many arrays [3] always subtract 1
2. **Objects**: Explains a singular item in depth (key value pairs)
   1. Var person = {
      1. Name: “Steve”, (String)
      2. Age: 21, (number)
      3. hairColor: “brown” (String)

**}**

Const addition

**SHORT CUTS AND COMMENTS**

**Shortcut console.log( ) Type in CLG and press Tab**

**Use Turbo Console Log plugin or Turbo JS**

**Highlight Matching Tag**

**Bracket Pair Colorizer**

**Indent rainbow**

// **comments** single row

**Shortcut ctrl + /**

/\* all

\*/