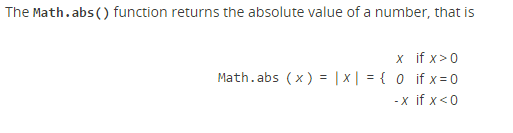
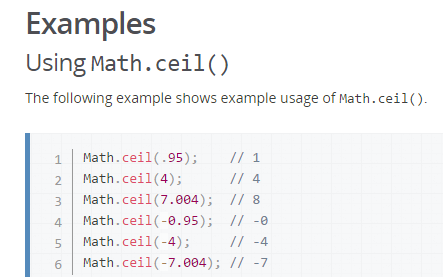
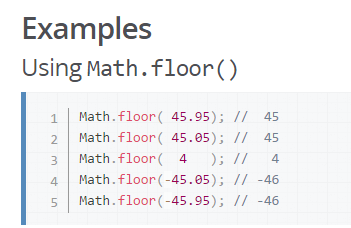
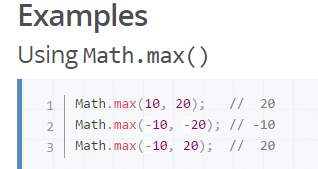
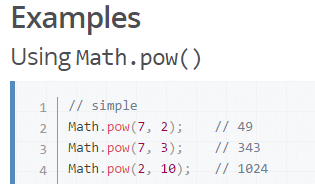
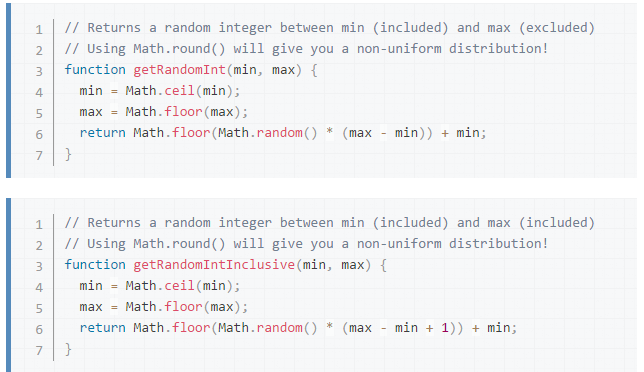
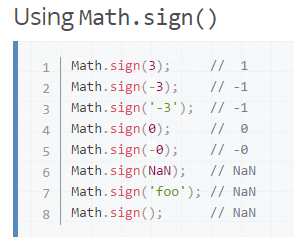
**Math Functions**

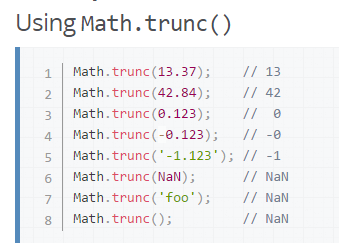
* Math.PI 🡪 3.14
* 
* 
* 
* Math.log10(x) & Math.log2(x)
* 
* Math.min()
* 
* Math.random()



* Math.round()
* Math.sign() tells the sign of the number



* Math.trunc()



**Difference between var and let**

The difference is scoping. var is scoped to the nearest function block and let is scoped to the nearest *enclosing* block, which can be smaller than a function block. Both are global if not in a block

{

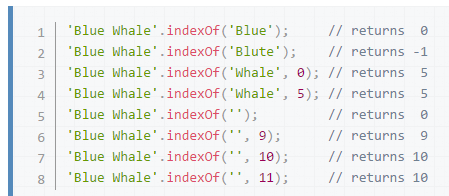
let a = 123;

};

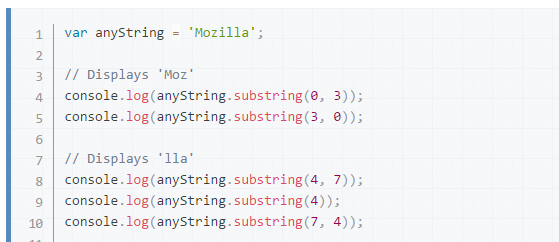
console.log(a); // ReferenceError: a is not defined

**String Functions**

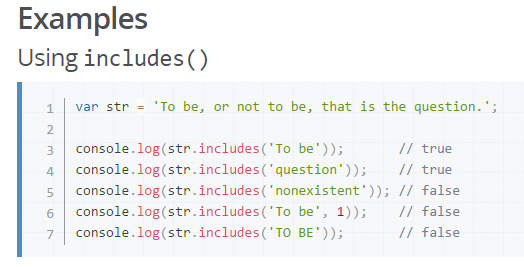
1. String.length
2. The **indexOf()** method returns the index within the calling [String](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String) object of the first occurrence of the specified value, starting the search at fromIndex. Returns -1 if the value is not found.



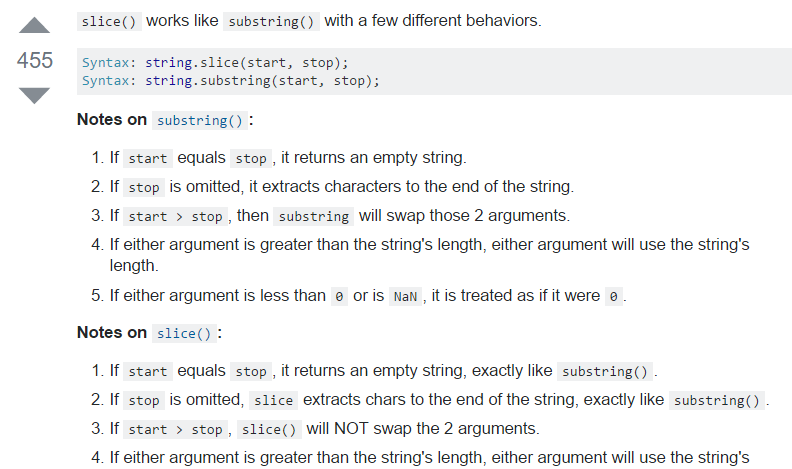
1. The **substring()** method returns a subset of a string between one index and another, or through the end of the string.

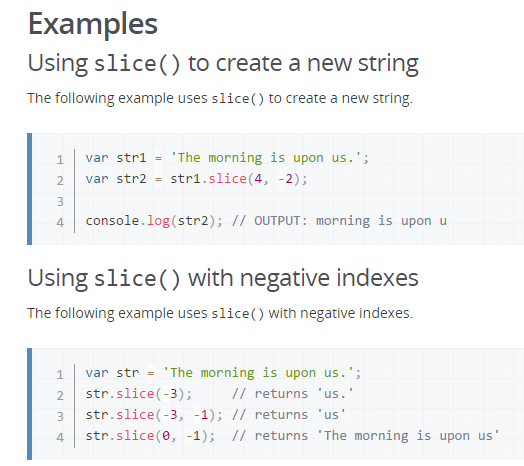


1. The **includes()** method determines whether one string may be found within another string, returning true or false as appropriate.



1. The **endsWith()** method determines whether a string ends with the characters of another string, returning true or false as appropriate
2. The **split()** method splits a [String](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String) object into an array of strings by separating the string into substrings.
3. The **slice()** method extracts a section of a string and returns a new string.





1. The **toLowerCase()** method returns the calling string value converted to lower case.
2. The **toString()** method returns a string representing the specified object.
3. The **toUpperCase()** method returns the calling string value converted to upper case.
4. The **trim()** method removes whitespace from both ends of a string. Whitespace in this context is all the whitespace characters
5. The **valueOf()** method returns the primitive value of a [String](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String) object.

**Ternary Operator**

function max(num1, num2){

var result= (num1 > num2? num1: num2);

return result

}

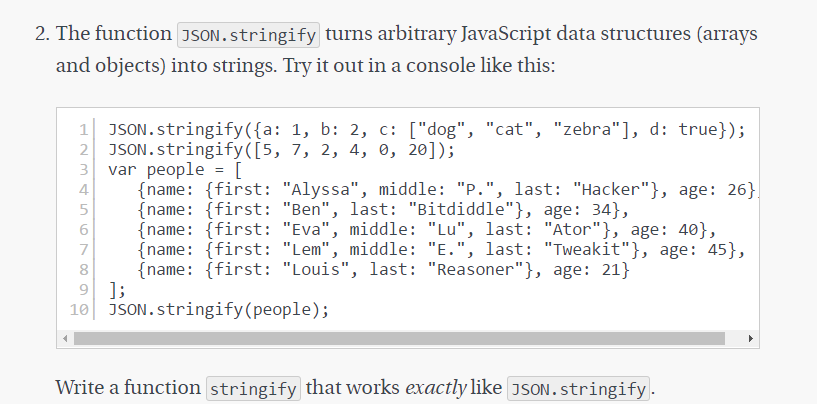
It turns out that we can write logical and and logical or in terms of each other and logical not using [De Morgan's Laws](https://en.wikipedia.org/wiki/De_Morgan%27s_laws).

* Write a function or that works like ||, but only uses ! and &&.
* Write a function and that works like &&, but only uses ! and ||.

W1d3

* Let's pretend for a moment that JavaScript does not have the addition operator + -- instead, it comes with two functions called inc and dec that perform increment and decrement respectively:
* Write a function called countChars that accepts two parameters: a string and a character. This function should return a number representing the number of times that the character appears in string. To access the first element of a string, you can use the following syntax:

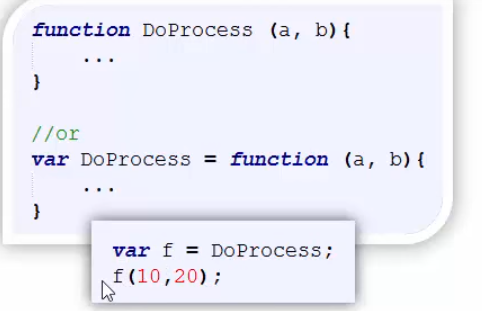
W2D2



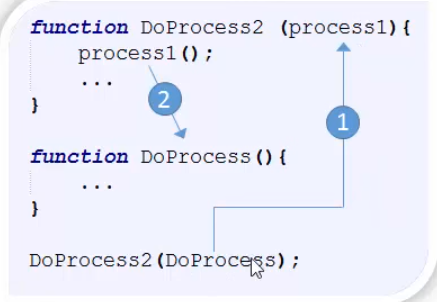
Functional expressions, functional literals

Anonymous functions

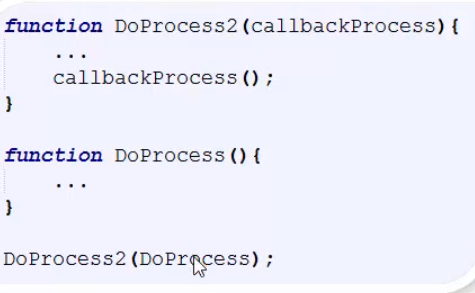
* Assigning function to another variable



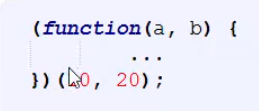
* Function passed as argument



* Callback



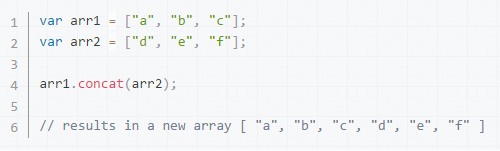
* Can invoke itself, self-executing anonymous functions



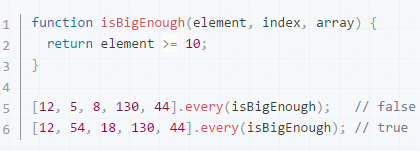
* Arguments 🡪 built in keyword which holds all the number of variables passed as arguments to a function.

Arrays

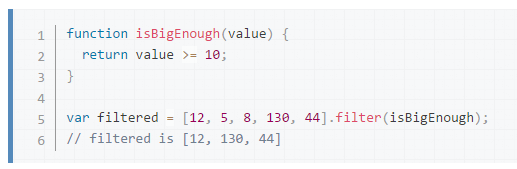
* The **concat()** method is used to merge two or more arrays. This method does not change the existing arrays, but instead returns a new array.



* The **every()** method tests whether all elements in the array pass the test implemented by the provided function.



* The **filter()** method creates a new array with all elements that pass the test implemented by the provided function.



function filter(arr,func){

var result =[];

each(arr,function(i){

if(func(i)){

result.push(i);

}

});

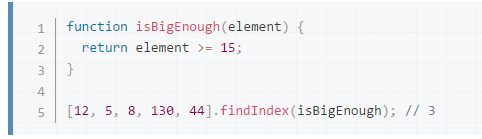
return result;

}

filter([1,2,3,4,5],function(r){ return r%2===0})

The **find()** method returns a **value** of the first element in the array that satisfies the provided testing function. Otherwise [undefined](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/undefined) is returned.

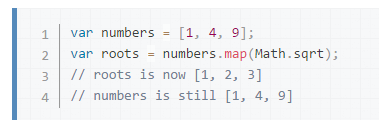
* The **findIndex()** method returns an **index** of the first element in the array that satisfies the provided testing function. Otherwise -1 is returned.

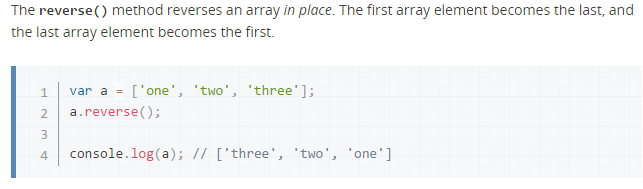


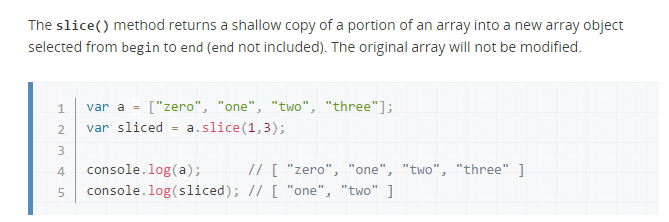
* The **forEach()** method executes a provided function once per array element.
* The **includes()** method determines whether an array includes a certain element, returning true or false as appropriate.
* The **indexOf()** method returns the first index at which a given element can be found in the array, or -1 if it is not present.



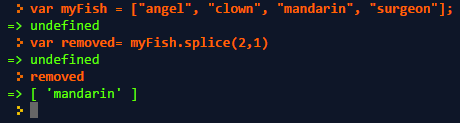
* The **map()** method creates a new array with the results of calling a provided function on every element in this array.



* Array push and unshift returns the current length of the array after performing their respective action
* 



* Arrays. splice(2,1) returns the deleted item in array (along with app Q11)



* The **splice()** method changes the content of an array by removing existing elements and/or adding new elements.



* **var min=Math.min.apply(null,numbers);**
* In order to check if a number is integer without fractional values. Use condition number%1 ===0 or not. Or Number. isInteger(num);

Date 21st Nov

(function() {

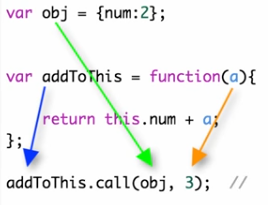
var a = b = 5;

})();

console.log(b);

answer: 5

* Call

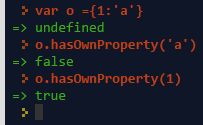


* Apply



Lessons learned (App academy)

* Object.hasOwnProperty() , only checks if keys are present or not. Not the values



* Please check Q12 of app academy solution. Most common letter to understand about objects
* Difference between for Each and map functions

Foreach

function dashString(num){

var array = num.toString().split("");

return array.forEach(function(letter, index){

return letter;

});

}

dashString(203)

//map

function dashString(num){

var array = num.toString().split("");

return array.map(function(letter, index){

return letter;

});

}

dashString(203)

* Slice with string return a string, slice with array returns array.
* Doubt in scramble program
* Prime factors

Why do we check up to the square root of a prime number to determine if it is prime?

Answer: If a number n is not a prime, it can be factored into two factors a and b:

n = a\*b

If both a and b were greater than the square root of n, a\*b would be greater than n. So at least one of those factors must be less than or equal to the square root of n, and to check if n is prime, we only need to test for factors less than or equal to the square root.

* Longest palindrome substring ( lesser complexity)
* To get the ascii value of a character

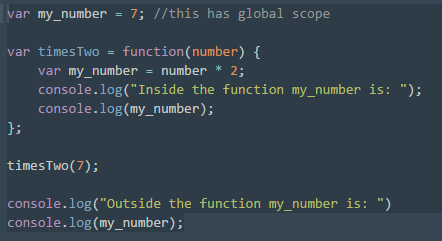
**'ABC'.charCodeAt(0)**

* + 65

Var result = String.fromCharCode(65);

* Foreach loop : does not return anything

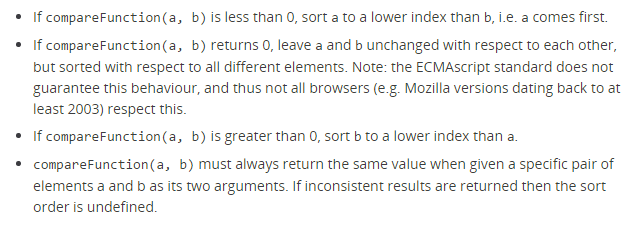
Codeacademy

* confirm(“how are you ?”) if ok then returns true else false
* By using == you check if something is equal to something else. This is not strict
* 3==='3'//false
* 3==="3"//false
* "3"==3//true
* 3===3//true
* 
* Splice() can be applied only to array not to string, it returns the removed item
* Slice() can be applied to both string and array , it return a new array with removed letters
* Mean , median and mode

Mean: average of the array

Median: sort the array, if odd number of elements, select middle number, if even take the avg of middle two number

Mode: more number of frequency

* Array.sort(function(a,b){ return a-b}); : The **sort()** method sorts the elements of an array in place and returns the array. The sort is not necessarily [stable](https://en.wikipedia.org/wiki/Sorting_algorithm#Stability). The default sort order is according to string Unicode code points.
* 

Coderbyte studies

* + 1. “hello”.replace(“e”,”o”); 🡺 “hollo”
* In objects 🡪 obj.name and obj[name] difference;
* **Arrays function forEach()**
* **“abc”.charAt(0) 🡪 “a”**