* The var keyword is used in all JavaScript code from 1995 to 2015.
* The let and const keywords were added to JavaScript in 2015.
* If you want your code to run in older browser, you must use var.

Let

* Variables defined with let cannot be Redeclared.
* Variables defined with let must be Declared before use.
* Variables defined with let have Block Scope.

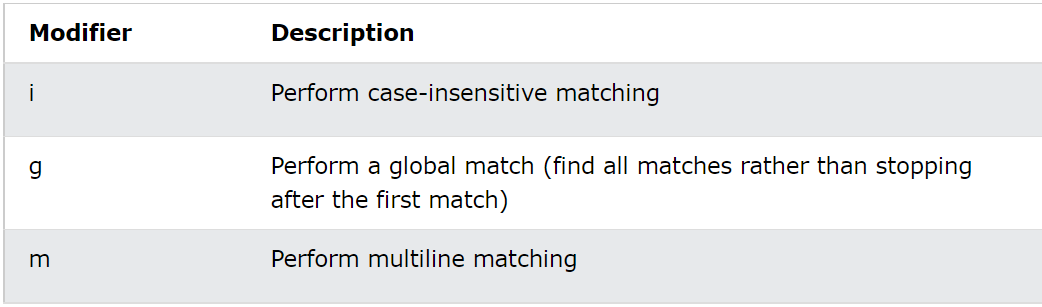
const

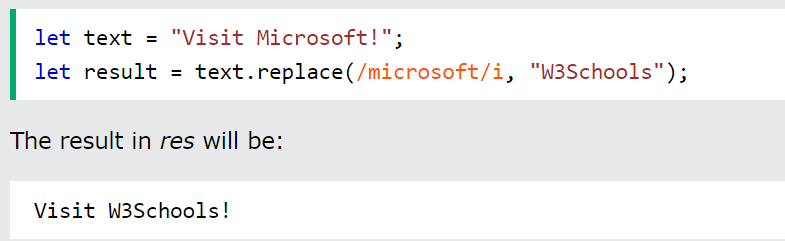
* Variables defined with const cannot be Redeclared.
* Variables defined with const cannot be Reassigned.
* Variables defined with const have Block Scope.

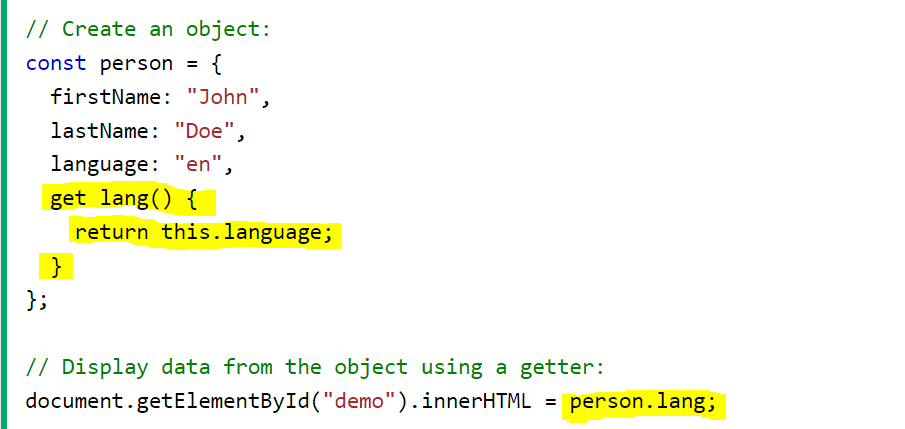
For example we have an object like toCelcius(f)=> (5/9)\*(f-32);

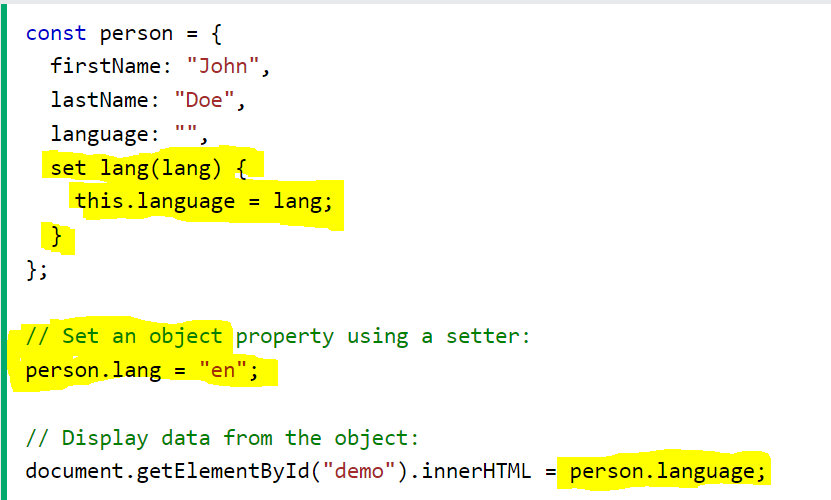
* toCelsius refers to the function object, and toCelsius() refers to the function result.
* The charAt() method returns the character at a specified index (position) in a string
* The charCodeAt() method returns the unicode of the character at a specified index in a string
* The search() method cannot take a second start position argument.
* The indexOf() method cannot take powerful search values (regular expressions).
* Comparin two JS objects always returns false
* toFixed() returns a string, with the number written with a specified number of decimals. It is perfect for working with money!
* If you use named indexes, JavaScript will redefine the array to an object. After that, some array methods and properties will produce **incorrect results**.
* The join() method also joins all array elements into a string.
* shift() method is equivalent to popping, but working on the first element instead of the last.
* The unshift() method adds a new element to an array (at the beginning), and "unshifts" older elements:
* Array elements can be deleted using the JavaScript operator delete.
* Using delete leaves undefined holes in the array.
* Use pop() or shift() instead.

RegEx





**Getter and Setter**



**Prototype:**

The JavaScript prototype property allows you to add new properties to object constructors.

EX: Person.prototype.nationality = "English";

* The call() method takes arguments **separately**.



* The apply() method takes arguments as an **array**.



* With the bind() method, an object can borrow a method from another object.

