ES6

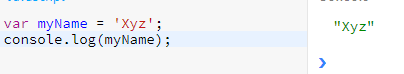
1. ‘const’ and ‘let’ keyword

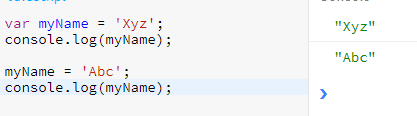
‘const’ enables you to define constants.

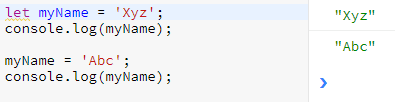
‘let’ enables you to define variables.

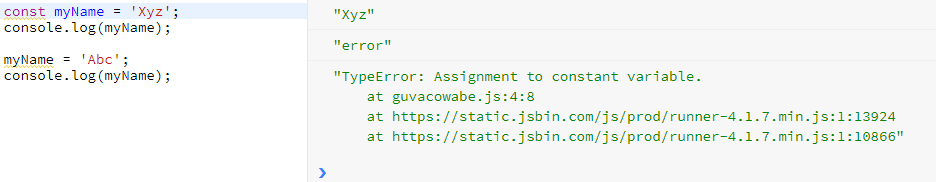
In JavaScript we define variables using ‘var’ keyword, which have function scope and are hoisted to the top. It means that a variable can we used before it has been declared.

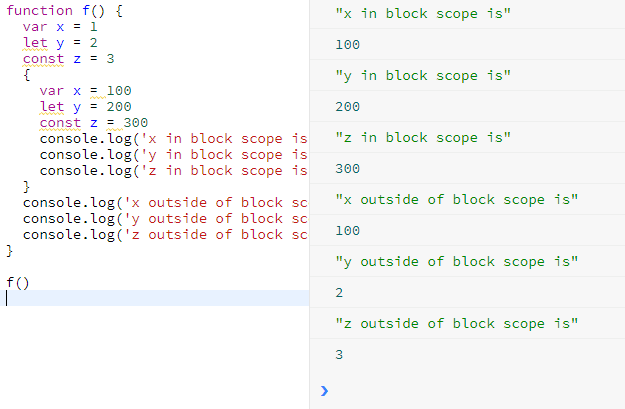
‘let’ and ‘const’ have a block scope (surrounded by {}) and cannot be used before declaration.











1. Array helper functions

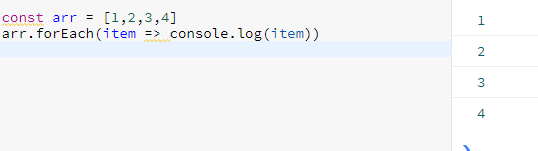
These help in data manipulation. There are different types of Array helper functions/methods in es6:

* forEach

The ‘forEach()’ method executes a provided function once per array element.

Use case: Iterate through an array

‘forEach()’ is a major upgrade over the ‘for’ loop

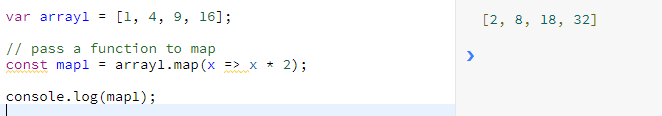


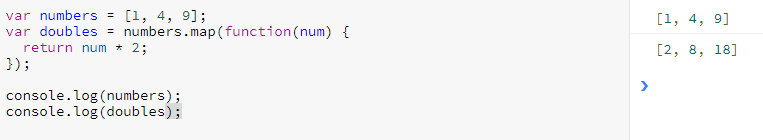


* map

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

Use case: parse and return an array of objects that contains a additional new property.



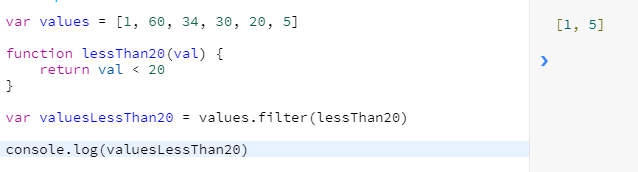


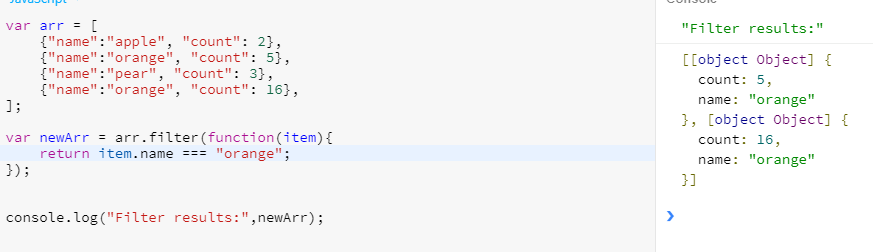


* filter

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

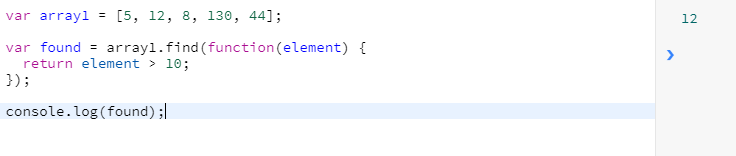
Filter return array based on the Boolean of the comparison.

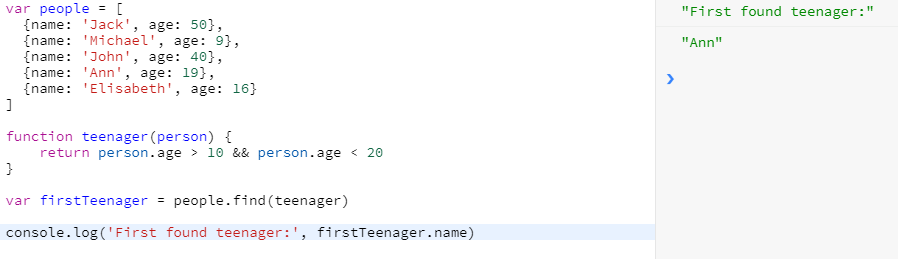




* find

The ‘find()’ method returns the value of the first element in the array that satisfies the provided testing function. Otherwise undefined is returned.

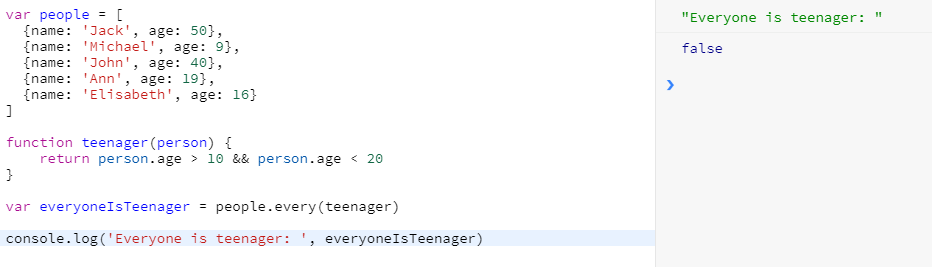


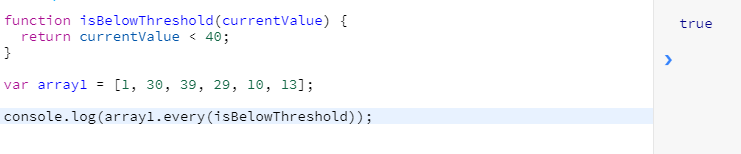


* every

The ‘every()’ method tests whether all elements in the array pass the test implemented by the provided function, which should return true or false

This method returns ‘true’ for any condition put on an empty array.

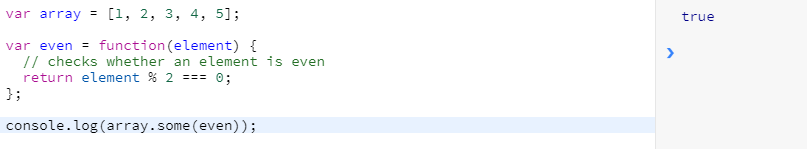




* some

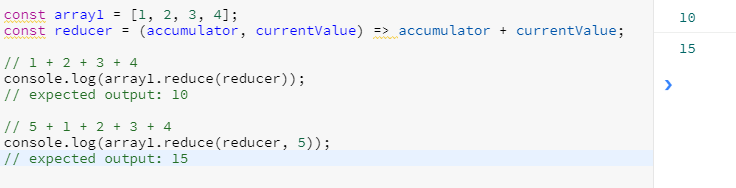
The ‘some()’ method tests whether at least one element in the array passes the test implemented by the provided function, which should return true or false

This method returns ‘false’ for any condition put on an empty array.



* reduce

The ‘reduce()’ method executes a reducer function (that you provide) on each member of the array resulting in a single output value.



1. Arrow functions

Arrow functions are also called as ‘fat arrow functions’

They have different syntax for creating JS functions.

Normal JS function looks like:

function myFnc(){

…..

}

Arrow function looks like:

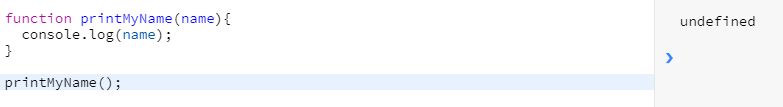
const myFnc = () => {

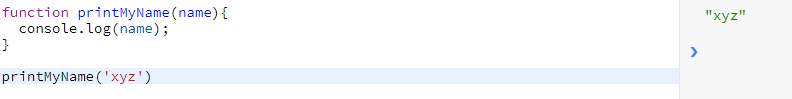
…..

}

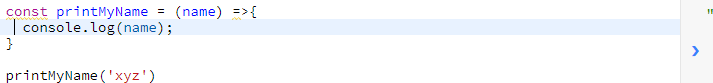
Arrow function solves a lot of the issues you often had with the ‘this’ keyword in JS.

Let’s see example for normal function

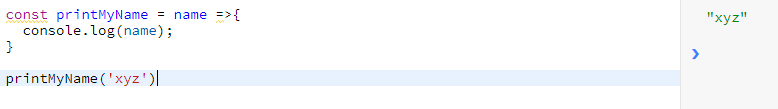




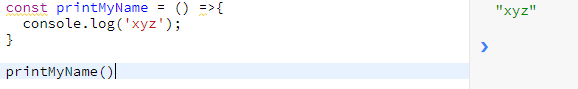
Now, arrow function



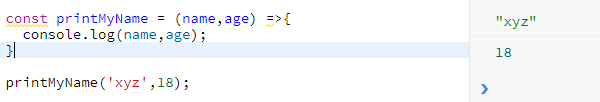
If you have only 1 argument

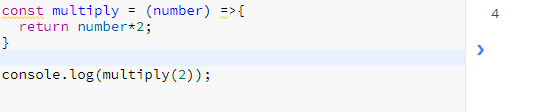


If you had a function which receives no arguments, you then need to pass an empty pair of parenthesis



If you have more than one argument



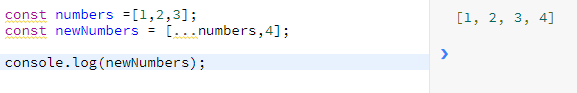


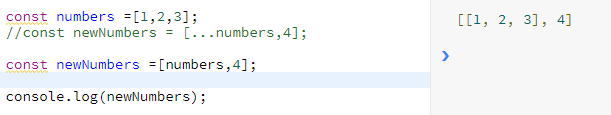
1. Rest and Spread operators (… dots)

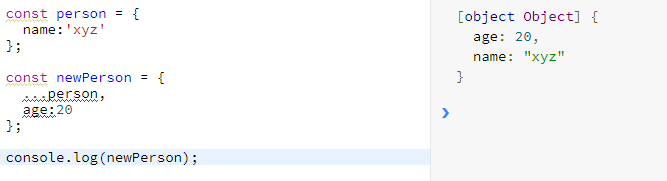
Spread - It allows to split an array to single arguments which are passed to the function as separate arguments.

const newArray = […oldArray,1,2]

const newObject = {…oldObject, newProp:5}

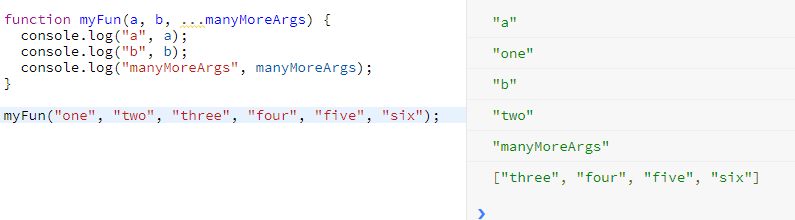






Rest – It allows us to represent an indefinite number of arguments as an array.

A function's last parameter can be prefixed with ‘...’ which will cause all remaining (user supplied) arguments to be placed within a "standard" javascript array. Only the last parameter can be a "rest parameter".

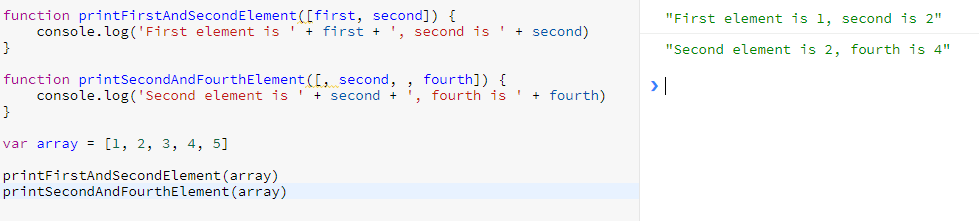


1. Destructuring

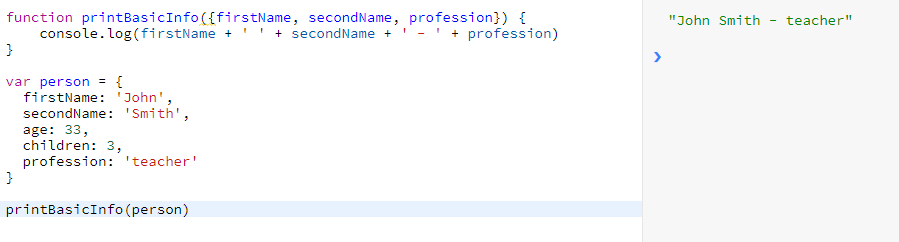
Destructuring allows you to easily extract elements or object properties and store them in variables.

Destructuring allows you to pull out single elements or properties and store them in variables for arrays and objects

* Of array



* Of object



1. Promises

A ‘Promise’ is an object that is used as a placeholder for the eventual results of a deferred (and possibly asynchronous) computation.

Promise has two channels: the first for results, the second for potential errors. To get the result, you provide the callback function as the ‘then’ function parameter. To handle errors, you provide the callback function as the ‘catch’ function parameter.

