Module 3 – Vanilla JavaScript: String, Booleans, Arrays and Objects

# Lab Objectives

To be able to use functions to manipulate strings

What is meant by a Boolean value and their uses

How to add to, remove from and loop through an array

Experience constructing and using Objects

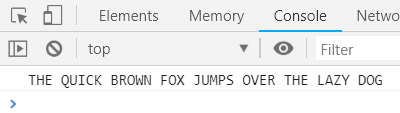
String Manipulation

Strings in JavaScript can be manipulated through a large number of existing functions. They allow you to do things such as get the character at a certain position, transform a string to upper case, and slice a string to only get the characters between two positions.

Many of these functions are easy to use, an example of transforming a string to upper case can be found below:

let exampleString = 'The quick brown fox jumps over the lazy dog';

console.log(exampleString.toUpperCase());

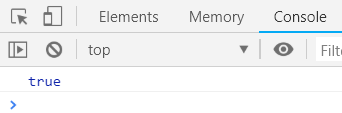


Task: Using the same string, use a string manipulation function to have only ‘quick brown fox’ printed in the console, also try using the search function to find out at which position the word brown starts

Task: Try some methods to compare strings such as startsWith, endsWith, includes etc. A simple example is shown below

let exampleString = 'The quick brown fox jumps over the lazy dog';

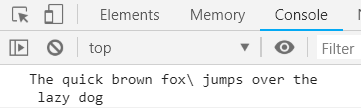
console.log(exampleString.includes('brown'));



You can also include some simple formatting in strings to allow new lines, and for special characters such as quotation marks to be inserted, an example can be found below:

let exampleString = 'The quick brown fox\\ jumps over the \n lazy dog';

console.log(exampleString);



Task: Research the other escape characters available and practice their usage

Booleans

Boolean values are simply true or false, in coding these are mainly used in if statements. True and false are the usual Boolean values however any variable type can be used as a Boolean value in an if statement. An example of using a string as a Boolean value can be seen below:

let exampleString = '';

function testString(passedString) {

if (passedString) {

console.log('string equates to true');

} else {

console.log('string equates to false');

}

}

testString(exampleString);

exampleString = 'non empty string';

testString(exampleString);

Task: Experiment and see if you can make other variable types equal both true and false (hint: try numbers, unassigned, null etc.)

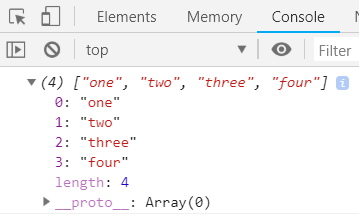
Arrays

Arrays are a sequence of values, arrays in JavaScript can change in size, unlike many other languages you do not need to specify a length. An example initialisation of a string array, and then dynamically adding to this can be found below

let exampleStringArray = ['one', 'two', 'three'];

exampleStringArray.push('four');

console.log(exampleStringArray);



Task: Find a way to loop through this array and print each element on a new line. (hint: research forEach function and recall fat arrow functions)

Task: take the following two arrays, find a way to join them, and then correctly sort the result from smallest to largest. (hint: research join and sort)

Objects

Objects are similar to arrays in that they can hold series of values, however in an object each value has a key that is used to access it. Objects can hold value of any type, and even functions. Below is an example initialisation of an object:

let exampleObject = {

"name": "John",

"age": 23,

"height": "5\'11\"",

"talk": () => console.log('talking'),

"favourite foods": ['pizza', 'lasagne']

}

As you can see, an object can contain a mix of types, functions and arrays. You can also add to this object dynamically using a line such as exampleObject.newValue = 5

**Task:** Create an object which describes a dog, have this contain arrays, functions, and strings, and use this to print the strings, and arrays, and also execute the methods

**Task:** Using the example above, create a function which takes in some values, and returns this object fully created for you, use this twice with different values to prove its functionality.