1. pass by value and pass by reference in JavaScript with data types?

Sol: Primitives are passed by value; Objects are passed by "copy of a reference”.

Specifically, when you pass an object (or array) you are (invisibly) passing a reference to that object, and it is possible to modify the *contents* of that object, **but if you attempt to overwrite** the reference it will not affect the copy of the reference held by the caller - i.e. the reference itself is passed by value:

2) arrays

JavaScript arrays are used to store multiple values in a single variable.

3) array methods indexOf, lastIndexOf, slice, splice, forEach, map, .length

**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 **lastIndexOf()** method returns the last index at which a given element can be found in the array, or -1 if it is not present. The array is searched backwards, starting at fromIndex.

The **slice()** method returns a shallow copy of a portion of an array into a new array object selected from begin to end (end not included). The original array will not be modified.

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

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

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

The **length** property represents the no of units of a array.

6) oops concepts in JavaScript

The two important principles with OOP in JavaScript are Object Creation patterns (**Encapsulation**) and Code Reuse patterns (**Inheritance**). When building applications, you create many objects, and there exist many ways for creating these objects: