1. Difference between copy by value and copy by reference.

**Copy by value**: In a primitive data-type when a variable is assigned a value we can imagine that a box is created in the memory. This box has a sticker attached to it i.e. the variable name. Inside the box the value assigned to the variable is stored.

**Copy by reference:** In case of a non-primitive data-type the values are not directly copied. When a non-primitive data-type is assigned a value a box is created with a sticker of the name of the data-type. However, the values it is assigned is not stored directly in the box. The language itself assigns a different memory location to store the data. The address of this memory location is stored in the box created.

1. Copy by value a composite data type.

Data type specify the kind of the data can be stored within a program. There are six data types in JS which are divided into three main categories as Primitive, Composite and Special data types. Numbers, Strings, and Boolean are of Primitive data type. Array, object, and Functions are of Composite data type. Undefined and Null are of Special data type. In Copy by value the data which variable hold is passed to another variable. Both the variables refers two different memory locations. There are 3 ways to copy by value for composite data types. Using the spread (…) operator, using the Object.assign() method, using the JSON.parse() methods.

Spread operator(...)

let a = [1, 2, 3];

let b = [...a];

Object.assign()

let a = [4, 5, 6];

let b = Object.assign([ ], a);

JSON.parse() and JSON.stringify()

let a = [7, 8, 9];

let b = JSON.parse(JSON.stringify(a));