1**.Difference Between call By value and call By reference**

Function can be invoked in two ways:

i)Call By value

ii)call By reference

these two ways are generally differentiated by the type of value passed to them as parameters.

**Call By Value - Call By reference**

i)While calling a function,we passed While calling a function,instead of passing values of

values of variables to it.such function variables,we pass address of varaiables(location)

are known as”Call By Values” to the function as”Call By Reference”

ii)In this method,the value of each in this methd,the address of actual variables in the

variable in calling function is copied calling function are copied into the dummy variable

into corresponding dummy varaiables of to the called function

the called function

iii)With this method,the change made to with this method,using addresses we would have an

the dummy variables in the called function access to the actual variables and hence we would

have no effect on the values of actual be able to manipulate them.

variables in the calling function

iv)The original variable is not modified The original variable gets modified on changes in

or changes in other variable other variables.

v)Actual and copied variables will be Actual and copied variables are created in the same

created in different memory location memory location

vi)Function arguments are always All objects intreact by reference

passed by value

3**)Dtatypes in javascript:**

i)There are six basic datatypes in Javscript .Which can be divided into three main categories.

\*Primitive(or Primary)

\*Composite(or Reference)

\*Trivial or Special data types

**ii)Primitive data types:**

\*string,number and booleans

**iii)composite data types:**

\*object,array,function

**iv) trivial or Special data types:**

\*Undefined and Null

**copy by value a composite data type:**

There are three ways to copy by value for composite data types:

i)Using the *spread (. . . )*operator

ii)Using the *object.assign()* method

iii)Using the *JSON.stringfy()* and *JSON.parse()* methods

**I)Using Spread:**

\*Spread operator allows an iterable to expands in palce where 0+ arguments are expected.

\*It is mostly used in the variable array where there is more than 1 values are expected.

\*Using spread opreator will clone your object.

\*copied variable value is changed but original variable value remain same.

**II)Using *Object.assign():***

\*The *object.assign()* method copies all enumerable own properties from one or more source objects to a target objects.

\*It returns the target object

**III) Using *JSON.parse()* and *JSON.sringify():***

\*The JSON object, available in all modern browser,has two useful methods to deal with JSON-formatted content: parse and stringify.

\*JSON.parse() takes a string and transforms it into a Javascript Object

\*JSON .stringify() takes a Javascript object and transform into a JSON string. \*Using JSON.parse() and JSON.stringify() for copy performs deep copy