```
// Operators

// Strict Comaprision Operators

// "===" and "!==" are the strict comaprision Operators
// that is used in javascript to Compare the Values
// and the data types both .

// Eg:
// let a  = 10;
// let b  = "10";

// console.log(a == b); true
// console.log(a != b); false
// console.log(a === b); false
// console.log(a !== b); true


// Difference between == and ===
```

// * "==" and "!=" Comapres only the values on both the Operands

// If the values are same it reuturns the

// boolean output as true or false.

// * "===" and "!==" Comapres the values and the dataype of

// values on both the Operands.

// If the values are same it reuturns the

// boolean output as true or false.

// Data types in javascript

// The type of a variable or objects created

// in javascript are classified using the datatypes.

// Data types are classified into two types .

```
// 1>primitive data types
// 2>Non - primitive data types

// 1>primitive data types
// Number
// String
// Boolean
// null
// undefined

// Number
//  is one of the primitive data types that is used to store
// decimal,integeral and long integersl values.

// Eg :
// let num = 100;
// console.log(typeof(num));


// String
//  is one of the primitive data types that is used to store
// Charecters and sequesnce of charecters.
// Srings are defined using "" or ''.

// Eg :
// let str = "Good Morning"
// console.log(typeof(str));

// Boolean
//  is one of the primitive data types that is used to store
// true or false values.
// In Javascript true means 1.
// In Javascript false means 0.

// Eg :
// let a = true;
// let b = false;
// console.log(a * b);
// null
// Null is a special data type that is used to be a
//  value and a datatype
// in javascript
// Null was invented as a bug in javascript but it was
// converted into a feauture.
// Null stands for empty value that can be re initialised by
// any kind of data type dynamically during execution.

// undefined
// undefined is a special data type that is used to be
// a value and a datatype
// in javascript
// undefined means that the variable has been declared but the
// value for the variable has not been initialised.
// Javascipt is a interpreted language the data type of
// a variable
// is based on a value that is passed by the user .
// So if the used dont assign any kind of value then the
// variable
```

// Will automatically stores the datatype as undefined.

// Eg :

// let a ;

// console.log(typeof(a));

// Arrays

// *arrays are the part of non primitive datatypes

// *Array is a collection of Homogoenous and Heterogeneous elements in javascript.

// We Can add diddfrent types of elements into a single array in javascript.

// Arrays are Dynamic in Size and allocation

// The elements in the array is added based on the index positions

// starting from zero.

// Array is mutable in nature

// *To make the array mutable in javascript we can use some array inbuilt methods.

// To create an array in javascript we can use [].

```
// Syntax :

// variableKeyword identifier = [value1,value2,value3,...]

// Eg :
// let arr = ["hi",true,21,...];

// List of array inbuilt methods
// arr.push
// arr.pop
// arr.unshift
// arr.shift
// arr.splice
// arr.slice
// arr.reverse
// arr.map
// arr.filter
// arr.reduce
// arr.toString
```

// arr.join

// arr.includes

// arr.lastIndexOf

// arr.indexOf

// arr.sort

// arr.push

// * push is a array inbuilt method that is used to add the number of elements into

// the array.

// *using push method we can add the elements at the last index position of an array.

// eg :

// let arr = ["hi",true,21];

// arr.push("hello",true);

// arr.pop

// * pop is a array inbuilt method that is used to remove the

// single element from the array.

```
// *using pop method we can remove single element
//  at the last index position of an array.

// Eg :
// let arr = ["hi",true,21];
// arr.pop();

// arr.unshift
// * unshift is a array inbuilt method that is used
// to add the number of elements into
// the array.
// *using unshift method we can add the elements
//  at the first index position of an array.

// eg :
// let arr = ["hi",true,21];
// arr.unshift("hello",true);

// arr.shift
// * shift is a array inbuilt method that is used to remove the
```

// single element from the array.

// *using shift method we can remove single element

// at the first index position of an array.

// Eg :

// let arr = ["hi",true,21];

// arr.shift();

// arr.splice

// splice is a array inbuilt method that is used to add or remover the element

// from any index position inside the array.

// splice method accept three arguments (start index,delete count , ...new Elements)

// Splide method removes number of elements from the specified start index

// and adds new value in the same place.

// Eg :

// let arr = ["hi",true,21];

// arr.splice(1,3,"true",false);

// arr.slice

// slice is a array inbuilt method that is used to create a sub array

// slice method cannot cannot affect the original array

// slice method takes two arguments (start index and delete count)

// it creates a copy of a new array from stating index position to end index position

// excluding the end index position

// eg :

// let arr = ["hi",true,21];

// arr.slice(1,3);

// arr.reverse

// reverse is a array inbuilt method that is uded to reverse the order of elements

// reverse method will affect the order of elements starting the last element

// in the first bucket

// eg :

// let arr = ["hi",true,21];

// arr.reverse();

// arr.toString

// toString is a Array inbuilt method that is used to convert the elements of a an

// Array into a sequence if strings.

// toString method cannot cannot affect the original array

// eg :

// let arr = ["hi",true,21];

// let a = arr.toString();

// arr.join

// join is a Array inbuilt method that is used to convert the elements of a an

// Array into a sequence if strings.

// join method cannot cannot affect the original array

// using join method we can specify the seperator between the

// array elements

// eg :

// let arr = ["hi",true,21];

// let a = arr.join(" ~ ");

// arr.includes

// includes is a array inbuilt method that returns only boolean values.

// includes method is used to search the existence of an element in

// the specified list of array

// if the element is found it will return true elese it will return false.

// includes method cannot cannot affect the original array

// eg :

// let arr = ["hi",true,21];

// let a = arr.includes(21);

// arr.indexOf

// indexOf is a array inbuilt method that is used to return the

// index position of a specified element in the array

// index of uses two arguments (search element,start index)

// it will search for the specified element from the starting from first index

// position .

// or it will search for the specified element from the starting inedx position

// indexOf method cannot cannot affect the original array

// eg :

// let arr = ["hi",true,21];

// let a = arr.indexOf(21);

// arr.lastIndexOf

// lastIndexOf is a array inbuilt method that is used to return the

// index position of a specified element in the array

// index of uses two arguments (search element,start index)

// it will search for the specified element from the starting from last index

// position .

// or it will search for the specified element from the starting inedx position

// lastIndexOf method cannot cannot affect the original array

// eg :

// let arr = ["hi",true,21];

// let a = arr.lastIndexOf(21);

// arr.sort

// arr.map

// arr.filter

// arr.reduce