**In JS, we have 7 primitive data types :**

1. Number (For Integer/Float)
2. String
3. Boolean
4. Undefined (whose value is not defined yet)
5. Null (also known as empty variable)
6. Symbol (ES2015) Unique whose value cannot be changed ( not of any work as of now))
7. BigInt (ES2020) (Large Integer values which cannot be hold by numbers)

**In JS, we don't need to define the type of variable, it will take the type from values.**

let a = 24; so here it will automatically become a number and if we will try to check its type it will give us as number, we can check the type with the following command : -

**Console.log(typeof a)**; here it will print number.

In JS we can also assign a variable to other types of values like we can do a="Vansh" now a is string.

We have three ways to define variables in JS :- **let, var, const**

**Const** : We use const for constant values like dob which can't be change later and we can't leave it unassigned it should be assigned at the time of declaration.

**Let&Var** : let and var are same as of each other the only big difference is let is block scoped and var is function scoped.

x\*\*y gives us x ki power y ex : - **2\*\*3 = 2\*2\*2**;

We have **template literals in JS** that allows us to create a string using variable and custom text much easier, we need to use back ticks for that.

Ex : let age=30;

Let ageString = `I'm ${age} years old`; so here we can add any variable using ${};

**Type Conversion and Type Coercion**

TO convert a type we can do : - const value = "1991";

const valueInNumber = **Number**(value);

So here value will be converted in number we can do same vice versa also like

value = **String**(valueInNumber);

it will convert number into string .

**Coercion** is when JS automatically convert one type to other like:

const value = 'Hi, I am '+ 30 + ' year old !';

so here JS will automatically convert number as 30 into string and will give us output as single string, so whenever we perform + on a number and a string, **number will be auto converted** into string. **If we will use (-,\*,/) in place of '+' then string will be converted into a number**.

Ex :- **value = '30' - 5; so here output will be 25, same for \* and /**

**Truthy and Falsy :**

So we have 5 falsy values in JS which conversion will give us false else all are truthy.

**5 falsy values are : 0, '', undefined, null, NaN ;**

**Else everything will give use true, like 1,'ac',{}.**

**== vs === :-**

So in JS we have two equality operators, one is == it checks only the value, another is === it check value as well as data type.

For ex : const value =18;

If(value=='18') -> It will give us true as 18 is equals to 18 anyhow.

If(value === '18') -> It will give us false as 18 is equals to 18, but one is string and another is number.

**Switch Statement :**

So the switch statement is used to check various conditions like days in the week :

const day = 'Monday'

Switch (day)

Case 'Monday' :

Console.log("Today is Monday");

Break;

Case 'Tuesday':

Console.log("Today is Tuesday");

Break;

Case 'Wednesday':

Case 'Thursday':

Console.log("Mid-week day, please focus on your coding part");

Break;

Default:

Console.log("the day is wrong");

**If we won't give the break after the case then it will execute the next case until it won't find the break;**

**The default is something, which will execute if all cases that we mentioned got failed at that case the default will be executed.**

**Expression vs Statement:**

**Expression :** Expression is something that produces some values. One Expression can have **multiple expression**s.

Ex: 1+2 = so this 1+2 will produce a value 3 here 1,2,1+2 this all three can be classified as different expressions or if we say 1 so it is producing a value 1. so any source of code that produces some value that is known as Expression.

**Statement** : Statement is a block of code that is executed by any language to do some functionality. Statement can have expression in it.

Ex: let number = 5; so here this complete line is a statement but 5 is an expression as it is producing a value 5.

**Ternary Operator or Conditional Operator :**

A ternary operator is an short way to write if else statement .

Let age =18;

Return age>=18 ? 10 : 5; so what this piece of code will do is, it will return the 10 if age is greater than or equals to 18 else will return 5.

**Strict Mode :**

We use strict mode in JavaScript to avoid some silly mistakes and avoid introducing the bugs. Like if we are doing some mistakes like if we are writing a wrong variable name then in console it will show us that we have used a wrong variable and will be helpful to identify that mistake and avoid the bug.

**Syntax :** In the very first line of the script file we need to just write 'user strict'; that's it.

**Exp :** let drivers = false; if(true) driver = true; so here we are expecting the drivers should become true but as we have used the wrong spell it will never become true and it will affect our functionality. So in this case if we will use strict mode it will give us error in console about this. Another test case for this is if we are using any reserved keyword for any variable name that it will show us the error in console.

**Function :**

**Syntax :** function sum(num1, num2){ return num1 + num2; } In js for arguments we don't need to specify the data type and also same for return type also. If we want to return anything or not we can do, no need to specify anything.