Node JS : Node JS is run time environment for JavaScript. Before Node JS JavaScript was known as

Client Side scripting language. If we want to develop any server side technologies we were depending upon others language like Java, python, php etc.

Node JS provided lot of pre defined modules (collection of function or classes etc).

Which help to create server side technology using node js we can connect to any database like mysql or mongo db etc. we can create rest api, we can create security related programs etc.

Server side technology

Java

Python

Php

Asp.net

Node JS

After node js we can say JavaScript also known as client side as well as server side scripting language.

Before Node JS if we want to run any JavaScript program we were using html code.

Ie that javascript may internal or external.

Using Node JS we can run JavaScript program using command prompt or terminal html code not required.

But in Node JS we can't use BOM and DOM.

console.log("Welcome to Node JS");

create the folder with name as NodeJs and TypeScript

please create the file with sample.js in VSCode

```
function sayHello(name){
   return "Welcome to node js "+name;
}
var result = sayHello("Akash");
console.log(result);
```

open any terminal ie vs code terminal or external terminal

and run the command as

node sample.js

TypeScript

TypeScript is a script language which is also known as super set of JavaScript.

JavaScript is known as loosely data type scripting language.

In TypeScript we can use data types (strict data types).

var salary=12000; // it is consider as number type.

salary=”120000abc” // it is consider as string type

salary = true;

In Typescript

var n:number=100;

n=”Ravi”; error

But Browser can’t understand TypeScript file we need to Convert Typescript to JavaScript.

We need to convert ts to js with help of transpiler

tsc (Typescript compiler) : it will help to convert ts to js.

Node js provided npm (node package manager)

Which help to download external node js modules.

npm --version

if we want to install any external module using npm command

npm install -g modulename -g global

npm install -g typescript

or

npm install -g typescript --force

please create the file datatype.ts

```
var n:number =10;
//n="Akash";
var m:number=10.10;
var fname:string ="Raj Deep";
var result1:boolean = true;
var msg:any="Hello"              // this variable can hold any types of value.
msg=true;
msg=100;
console.log(n);
console.log(m)
console.log(result1)
console.log(fname)
console.log(msg)
```

first convet ts to js using command as

tsc datatype.ts

node datatype.js

ES5 and ES6 Features using TypeScript.

From ES6 onward to declare the variable we use var, let and const keyword.

Using var keyword we can re-declare same variable once again with same value or different value.

```
var a:number=10;                // declare and assign the value
a=20;                           // initialization
var a:number=30;                // re-declaration

let b:number=10;
b=20;
let b:number=30;                // b already declared.    Error

int a=10;                       // declare and assign the value
a=20;
a=30;
a=30;              // a variable already declared
```

using var we can re-declare same variable once again.

Using let we can't re-declare same variable once again.

But we can re-assign the value of let as well as var.

const we can't re-assign the value ie constant.

var variable global scope

let variable consider as local or block scope.

varLetAndConst.ts

```
var a:number=10;        // declare variable and assign the value
a=20;                   // change the value
a=40;
var a:number=30;        // re-declaration once again with different value.
let b:number =10;
b=20;
b=30;
//let b:number =30;         // we can't re-declaration
const c:number =10;
//c=20;             // can't reassign or change

for(var i=0;i<1000;i++){

}
console.log("Value of i is "+i);
for(let j=0;j<10000;j++){
    console.log(j);
}
//console.log("Value of j is "+j)
```

JavaScript function can return or it can't return

If return it can return number or b oolean or string etc.