HTML/HTML5

CSS/CSS3

JavaScript

Bootstrap

jQuery

JavaScript : ES5 Features

ES6 ECMA Script Features

Angular 1.x Angular 2.x/4.x/5.x/6.x/7.x/8.x

Angular is a framework. Framework is implementation of design pattern.

Angular 1.x base upon MVC Design pattern.

html/css/JavaScript

Angular 2.x to 8.xt upon Component base architecture

html/css/TypeScript

React JS : Library function

TypeScript :

Node JS : Node JS is a runtime environment for JavaScript library or framework like In Java JRE.

With the help of Node js we can write Server Side JavaScript coding.

RestFull Service Using JavaScript

MEAN Stack : Mongo DB/MySQL Express Angular Node JS

MERN Stack React JS

MEVN Stack Vue JS

TypeScript : TypeScript is a super set of JavaScript scripting language.

TypeScript support all features of ES6 or ES7 or ES8.

TypeScript Features

1. var, let and const

2. datatypes like number, string, boolean, enum, array etc.

3. looping

while loop, do while loop or for loop

for each loop

for in loop

for of loop

4. functions

1.normal function

2. function with parameter with datatypes

3. function with return types

4. spread operator function

5. rest operator function

6. arrow functions

5. OOPs

a. class

b. constructor

c. interface

d. extends

e. implements

etc

6. modules like package in java

<script type="text/JavaScript" src="abc.js"></script>

var and let

let keyword is use to declare the local scope or block scope variable where var keyword is use to declare global scope.

using var keyword we can do re-declaration but not with let keyword.

int a=10;

int a=20;

var a=10;

var a=20;

npm : node package manager:

This tool or command is use to download the dependencies.

npm install typescript

C:\Users\Akash\AppData\Roaming\npm

abc.ts

console.log("Welcome to TypeScript Program");

tsc abc.ts

node abc.js

abc.ts

let a =10;

let b =20;

let sum = a+b;

console.log("Sum is "+sum);

abc.ts

/\*function display(name){

if(name=="Ravi")

{

let msg = "Welcome Ravi";

}else {

let msg ="Welcome Unknown"

}

return msg;

}

console.log(display("Ravi"));

\*/

for(let i=0;i<10000;i++){

}

console.log(i);

abc.ts

var a=10;

var a =20;

//let b = 30;

//let b =40;

const c =30;

c = 50;

var a=10;

var a =20;

//let b = 30;

//let b =40;

const c =30;

c = 50;

DataTypes

var a=20;

let b:number=20;

let c:number;

let fname:string;

let result:boolean;

let msg:any;

//a="Ravi";

//b = "Mahesh";

c=30;

fname="Ravi";

result=true;

msg=30;

msg="Welcome";

Array

var num =[10,20,"Ravi",true,40];

var num:Array<any>=[10,"Ravi"];

var num1:Array<string>=["Ravi","Mahesh","Ramu"];

var num2:Array<number>=[10,20,30,40,50];

console.log(num[0]);

Looping

var num:Array<number>=[10,20,30,40,50,60];

console.log("for loop");

for(let i=0;i<num.length;i++) {

console.log(num[i]);

}

console.log("for in loop");

for(let i in num) {

console.log(i+"---"+num[i]);

}

console.log("for of loop");

for(let n of num) {

console.log(n);

}

console.log("using for each function");

num.forEach(function(val){

console.log(val);

})

console.log("using for each external function ");

num.forEach(dis);

function dis(val) {

console.log(val);

}

public void main(String...args) {

}

main();

main("a");

main("a","b");

String abc[]={}

main(abc)

function readValue1(a:any) {

}

function readValue2(a:number) {

}

function readValue3(num:Array<any>) {

}

function readValue4(...num1:Array<Array<number>>) { //Rest operator

/\*for(let n of num1){

console.log(n);

}\*/

console.log(num1);

}

readValue4();

readValue4(10);

readValue4(10,20,30);

//readValue4(1,"Ravi",15000);

let a:Array<number>=[1,2,3,4];

let b:Array<number>=[1,2,3,4];

let num:Array<Array<number>>=[a,b];

readValue4(num);

//readValue4(num);

//readValue4(...num,...num); //spread operator

Arrow functions :

Arrow function like a lambda expression in Java.

let num:Array<number>=[10,20,30,40];

num.forEach(display);

function display(value) {

console.log(value);

}

num.forEach(function(val) {

console.log(val);

})

num.forEach( (val) => console.log(val) );

let num:Array<number>=[10,20,30,40];

num.forEach(display);

function display(value) {

console.log(value);

}

num.forEach(function(val) {

console.log(val);

})

num.forEach( (val) => console.log(val) ); //arrow function

//ES6

class Employee {

empId;

empName;

empSalary;

/\*constructor() {

console.log("Object created");

}\*/

constructor(empId:number,empName:string,empSalary:number) {

console.log("Object created");

this.empId = empId;

this.empName = empName;

this.empSalary=empSalary;

}

displayEmpInfo():void {

console.log("Display empinfo");

console.log("emp id "+this.empId);

console.log("emp name "+this.empName);

console.log("emp salary "+this.empSalary);

}

}

//let emp = new Employee();

//emp.displayEmpInfo();

let emp = new Employee(100,"Ravi",14000);

emp.displayEmpInfo();

//ES6

class Employee {

constructor(public empId:number,public empName:string,public empSalary:number) {}

displayEmpInfo():void {

console.log("Display empinfo");

console.log("emp id "+this.empId);

console.log("emp name "+this.empName);

console.log("emp salary "+this.empSalary);

}

}

let emp = new Employee(100,"Ravi",14000);

emp.displayEmpInfo();

WADL : REST

WSDL : SOAP

//ES6

class Employee {

constructor(private empId:number,public empName:string,private empSalary:number) {}

displayEmpInfo():void {

console.log("Display empinfo");

console.log("emp id "+this.empId);

console.log("emp name "+this.empName);

console.log("emp salary "+this.empSalary);

}

}

let emp = new Employee(100,"Ravi",14000);

emp.displayEmpInfo();

console.log(emp.empName);

//ES6

class Employee {

constructor(private empId:number,public empName:string,private empSalary:number) {}

setEmpId(empId:number):void {

this.empId = empId;

}

getEmpId():number {

return this.empId;

}

}

let emp = new Employee(100,"Ravi",14000);

console.log(emp.getEmpId());

npm install -g @angular/cli

or

npm install @angular/cli

ng new demo-app