**TypeScript Demos:**

**Demo1:**

function greeter(person) {

return "Hello, " + person;

}

var user = "Jane User";

document.body.innerHTML = greeter(user);

**Demo2:**

function greeter(person: string) {

return "Hello, " + person;

}

var user = [0, 1, 2];

document.body.innerHTML = greeter(user); //Error as we are passing an array instead of String

**Demo3: - Interface**

**class Person{**

**name: string;**

**age: number;**

**}**

**function sayHello(p1: Person)**

**{**

**return p1.name + " " + p1.age;**

**}**

**var person1 = { name: "Hema", age: 18 };**

**document.body.innerHTML = "<h1>"+sayHello(person1)+"</h1>"**

interface Person {

firstName: string;

lastName: string;

}

function greeter(person: Person) {

return "Hello, " + person.firstName + " " + person.lastName;

}

var user = { firstName: "Jane", lastName: "User" };

document.body.innerHTML = greeter(user);

**Demo4: - class**

class Employee

{

id: number;

name: String;

desig: String;

constructor(x: number, y: string, z: string)

{

this.id = x;

this.name = y;

this.desig = z;

}

}

---------------------------

class Person{

name: string;

age: number;

constructor(n1: string)

{

this.name = n1;

this.age = 40;

}

}

function sayHello(p1: Person)

{

return p1.name + " " + p1.age;

}

var per1: Person;

per1 = new Person("Adhi");

document.body.innerHTML = "<h1>"+sayHello(per1)+"</h1>"

**Demo5: - class and Object**

class Employee

{

private id: number;

private name: String;

private desig: String;

constructor(x: number, y: string, z: string)

{

this.id = x;

this.name = y;

this.desig = z;

}

}

var e1: Employee;

e1 = new Employee(89, 'John', 'TL');

document.body.innerHTML = e1.id + " Name : " + e1.name;

class Employee

{

private id: number;

private name: String;

private desig: String;

constructor(x: number, y: string, z: string)

{

this.id = x;

this.name = y;

this.desig = z;

}

display(cnt:number):string

{

return "<h1>" + this.name + " Count = "+cnt+"</h1>";

}

}

var e1:Employee = new Employee(10,"Leena","TL");

document.body.innerHTML = e1.display(5);

**Demo6: - Inheritance – Constructor**

**class Person**

**{**

**name: string;**

**age: number;**

**}**

**class Student extends Person**

**{**

**marks: number[];**

**isPass: boolean;**

**constructor() {**

**super();**

**this.marks = [78, 56, 12, 34];**

**this.isPass = true;**

**}**

**}**

**var s1: Student = new Student();**

**document.body.innerHTML += "<h1> Name : " + s1.name + " Age : " + s1.age;**

**document.body.innerHTML +="<h1> Marks : "+ s1.marks + " Result : " + s1.isPass;**

**----------------------------------------**

class Person

{

name: String;

constructor( y: string) {

this.name = y;

}

}

class Employee extends Person

{

id: number;

desig: String;

constructor(x: number, y: string, z: string)

{

super(y);

this.id = x;

this.desig = z;

}

}

var p1: Person;

p1=new Person('Allen')

document.body.innerHTML = " Person : " + p1.name;

var e1: Employee;

e1 = new Employee(89, 'John', 'TL');

document.body.innerHTML +="<br/>"+ e1.id + " Name : " + e1.name;

**Demo7: - Getters and Setters using functions**

class Student{

private id: number;

private name: String;

constructor(x: number, y: string)

{

this.id = x;

this.name = y;

}

getId() { return this.id; }

getName() { return this.name;}

}

var s1: Student = new Student(89, "Tina");

document.body.innerHTML="<h1>"+s1.getId()+" "+s1.getName()+"</h1>"

**Demo7: - Typescript Getters and Setters**

class Student{

private \_id: number;

private \_name: string;

constructor(x: number, y: string)

{

this.\_id = x;

this.\_name = y;

}

get id():number { return this.\_id; }

set id(x) { this.\_id = x; }

get name():string { return this.\_name; }

set name(x) { this.\_name=x;}

}

var s1: Student = new Student(89, "Tina");

document.body.innerHTML="<h1>"+s1.id+" "+s1.name+"</h1>"

**Demo 8: Static**

class Account

{

static interestRate: number;

accNo: number;

constructor(no: number)

{

this.accNo = no;

}

public static getRate()

{

return Account.interestRate;

}

}

var a1: Account = new Account(101);

document.body.innerHTML ="<h1>"+ a1.accNo+"</h1>";

document.body.innerHTML ="<h1> Static value :"+ Account.getRate()+"</h1>";

**Demo 9 : Optional, Default and Varargs parameters**

**function display(name:string,age:number,loc?:string,sal?:number)**

**{**

**return name+" "+age+" "+loc+" "+sal;**

**}**

**document.body.innerHTML="<h1>"+display("Allen",23)+"</h1>";**

**document.body.innerHTML+="<h1>"+display("Allen",23,"CHN")+"</h1>";**

**document.body.innerHTML+="<h1>"+display("Allen",23,"CHN",78657)+"</h1>";**

**document.body.innerHTML+="<h1>"+display("Allen",23,78657)+"</h1>";**

**function display(name:string,age:number,loc:string="Delhi",sal:number)**

**{**

**return name+" "+age+" "+loc+" "+sal;**

**}**

**document.body.innerHTML="<h1>"+display("Allen",23)+"</h1>";**

**document.body.innerHTML+="<h1>"+display("Allen",23,"CHN")+"</h1>";**

**document.body.innerHTML+="<h1>"+display("Allen",23,"CHN",78657)+"</h1>";**

**document.body.innerHTML+="<h1>"+display("Allen",23,78657)+"</h1>";**

**----------------------**

function buildName(firstName: string, middleName?: string, lastName = "M", ...others:string[]) {

if (middleName !== undefined)

console.log("%s %s %s", firstName, middleName, lastName);

else

console.log("%s %s", firstName, lastName);

if(others.length > 0){

others.forEach(function(item){

console.log(item);

});

}

}

buildName("Karthik");

buildName("Ashik", "Mohammed", "Ibrahim");

buildName("Ashik", "Mohammed", "Ibrahim","Test1","Test2");

-------------------------------------------------------------------------------------------------------------------------------------------------

**Demo**

class Employee {

id: number;

name: String;

desig: String;

constructor(x: number, y: string, z: string) {

this.id = x;

this.name = y;

this.desig = z;

}

}

var data: Employee;

data = new Employee(10, "priya", "Software Eng");

document.body.innerHTML += "<h1>" + data.name +"</h1>";

document.body.innerHTML += "<h2>" + data.desig + "</h2>";