Function with optional parameters

===============================

function disp\_details(id:number,name:string,mail\_id?:string) {

console.log("ID:", id);

console.log("Name",name);

if(mail\_id!=undefined)

console.log("Email Id",mail\_id);

}

disp\_details(123,"John");

disp\_details(111,"mary","mary@xyz.com");

Rest Parameters - Rest parameters are similar to variable arguments in Java

=================

function addNumbers(...nums:number[]) {

var i;

var sum:number = 0;

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

sum = sum + nums[i];

}

console.log("sum of the numbers",sum)

}

addNumbers(1,2,3)

addNumbers(10,10,10,10,10)

Default Parameters

==================

function calculate\_discount(price:number,rate:number = 0.50) {

var discount = price \* rate;

console.log("Discount Amount: ",discount);

}

calculate\_discount(1000)

calculate\_discount(1000,0.30)

Anonymous Function

====================

- These functions are dynamically declared at runtime. Anonymous functions can accept inputs and return outputs, just as standard functions do.

var msg = function() {

return "hello world";

}

console.log(msg())

Anonymous function with parameters

===============================

var res = function(a:number,b:number) {

return a\*b;

};

console.log(res(12,2))

Function Recursion

=================

function factorial(number) {

if (number <= 0) { // termination case

return 1;

} else {

return (number \* factorial(number - 1)); // function invokes itself

}

};

console.log(factorial(6));

Lambda Functions – Arrow Functions 🡪 represents anonymous functions

var foo = (x:number)=>10 + x

console.log(foo(100))