TypeScript N Hemanth

### Day1

Variables, functions, arrays, iterating.

### Tasks

- List different types in typescript and their declarations  
- Define some functions  
 Function with parameters - with return types  
 Function with parameters - without return types  
 Function without parameters - without return types  
 Function without parameters - with return types  
- Define same functions for arrow functions.

- Define some arrays and iterate using different loops.

- Define an empty string array and then add values and remove values from the array.

- Create a function to check if a given year is a leap year or not that returns string “Leap Year” or “Not a Leap Year”

- Practice all inbuilt methods on string and arrays.

Q. List different types in typescript and their declarations ?

→ three types of type system is available in type scripting

1. Any
2. Built-in type
3. User type

Any : any is super type of all data types.it denotes a dynamic type. Using any type is equivalent to any type output.

Built-in type : total 6 data types are available in typescript .

1. Number : it can represent both frictional and integer type. It have 64 bit floating points.
2. String: it’s a series of collection of characters.
3. Boolean : it return true or false
4. Void : it represent non returning functions.
5. Null : represents an intentional absence of object value.
6. Undefined : denote the the values to all un-initialized variables.

**NOTE:** Both null and undefined are not used for variable reference instead it can assign as values to variables.

User type : User type data types are enums,classes,interfaces and arrays etc.

Q) Define some functions  
 Function with parameters - with return types  
 Function with parameters - without return types  
 Function without parameters - without return types  
 Function without parameters - with return types

/\*

- Define some functions

Function with parameters - with return types

Function with parameters - without return types

Function without parameters - without return types

Function without parameters - with return types

\*/

// function with parameters - with return types

function withParameters(num:number,num1:number,s:String) : String {

// body...

var s : String;

if (num1%num==0) {

// code...

s = 'true';

} else {

// code...

s = 'false';

}

return s;

}

var result = withParameters(2,4,'check');

console.log(result)

// Function with parameters - without return types

function withoutReturnType(num:number) {

// body...

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

// code...

console.log(i);

}

}

withoutReturnType(5);

//function without Parameters - without return types

function withoutParam() {

// body...

let firstcom : String = "duck married to buckless duck";

let secondcom : String = "buckless duck got divorced from buck duck";

console.log(firstcom+" omg "+secondcom);

}

withoutParam();

//Function without parameters - with return types

function wopwrt() : boolean {

// body...

var a : number = 25;

var b : number = 25;

if (a==b) {

// code...

return true;

} else {

// code...

return false;

}

}

var check\_result = wopwrt();

console.log(check\_result);

Q) Define same functions for arrow functions.

/\*

Arrow Function with parameters - with return types

Arrow Function with parameters - without return types

Arrow Function without parameters - without return types

Arrow Function without parameters - with return types

\*/

// Function with parameters - with return types

var multiplication = (num:number)=>{

for (var i = 0; i < 20; ++i) {

// code...

return (num\*i);

}

}

var res = multiplication(15);

console.log(res);

// Function with parameters - without return types

var details=(name :string,rollNo: number,age?:number)=>{

console.log("Name is:",name);

console.log("roll number is :",rollNo);

if(age!=undefined)

console.log("Age is",age);

}

console.log("Details of first student :");

details("vamsi",100,23);

console.log("Details of second student :");

details("harshit",101);

var caught = (thief : String)=>{

let name: String = "superman";

if (thief==name) {

// code...

console.log("thief name is superman");

} else {

// code...

console.log("thief escape");

}

}

caught("veerapan");

// Function without parameters - without return types

var prnt=()=>{

console.log( "Function without parameters - without return types");

}

prnt();

// Function without parameters - with return types

var rnt=()=>{

return "Function without parameters - with return types";

}

console.log(rnt());

Q) Define an empty string array and then add values and remove values from the array using (push and pop).

// Define an empty string array and then add values and remove values from the array using (push and pop)

var arr: String[] = ['h','e','m','a','n','t']

var lnth = arr.push('h');

console.log("length of array is ="+lnth);

console.log("String elements in array are:")

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

console.log(arr[i]);

}

var lett = arr.pop();

console.log("poped elemnts from array is = "+lett);

Q) Create a function to check if a given year is a leap year or not that returns string “Leap Year” or “Not a Leap Year.

function leaporNot(year:number) : boolean{

if ((year % 4 ==0) && (year % 100 !=0) || (year % 400 ==0)) {

// code...

return true;

} else {

// code...

return false;

}

}

var year : number = 1993;

var result = leaporNot(year);

if (result==true) {

// code...

console.log(year + " is a leap year.")

} else {

// code...

console.log(year + " is not a leap year.")

}

Q) Practice all inbuilt methods on string and arrays. ?

// Practice all inbuilt methods on string and arrays.

var str : String = "practise purpose";

var nameReg = /^[a-z ,.'-]+$/i;

//charAt() method

var length = str.length;

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

// code...

console.log(str.charAt(i));

}

//charCodeAt()

console.log("charCodeAt() methods values : "+str.charCodeAt(1));

//concat() method

var firstname = "hemanth";

var lastname = "nidamanuri";

var finalname ;

finalname = firstname.concat(lastname);

console.log(finalname);

//indexOf

console.log("index of 'purpose' in string is ="+str.indexOf("purpose"));

//lastIndexOf

console.log("lastindexof = "+str.lastIndexOf("purpose"));

//localcompare

console.log("localcompare = "+str.localeCompare("test"));

//match

console.log("match() method = "+ "00000".match(nameReg));

//replace()

var test : String = "hemants";

console.log("replace() replaces s with h in my name : "+test.replace("s","h"));