

## Typescript Tasks

### Day1

#### Variables :

Variable must declare before its use using var or let.

- 1.Variable names can alphabets and numerics.
2. Variables cannot begin with digit

#### Declaration of variables :

Variables in typescript can declare in different ways

- 1.Declare its type and value in one statement.

Ex var a : number = 10;

- 2.Declare its type but no value. In this case, the variable will be set to undefined.

Ex var a : number;

- 3.Declare its value but no type. The variable type will be set to the data type of the assigned value.

Ex var a = 10;

- 4.Declare neither value not type. In this case, the data type of the variable will be any and will be initialized to undefined.

Ex var a;

#### The Any type

The any data type is the super type of all types in TypeScript. It denotes a dynamic type. Using the any type is equivalent to opting out of type checking for a variable.

#### Built-in types

The following table illustrates all the built-in types in TypeScript –

Data type	Keyword	Description
Number	number	Double precision 64-bit floating point values. It can be used to represent both, integers and fractions.
String	string	Represents a sequence of Unicode characters
Boolean	boolean	Represents logical values, true and false
Void	void	Used on function return types to represent non-returning functions

Null	null	Represents an intentional absence of an object value.
Undefined	undefined	Denotes value given to all uninitialized variables

*null and undefined are not the same.* A variable initialized with undefined means that the variable has no value or object assigned to it while null means that the variable has been set to an object whose value is undefined.

## User-defined Types

User-defined types include Enumerations (enums), classes, interfaces, arrays, and tuple. These are discussed in detail in the later chapters.

### Function with parameters - with return types

```
function add(a:number,b:number) : number{
    return a+b;
}
var value:number = add(5,6);
console.log(value);
```

### Function with parameters - without return types

```
function display(a:number,b:number):void{
    console.log(a+b);
}
display(7,6);
```

### Function without parameters - without return types

```
function multiply():void{
    var a:number = 10;
    var b:number = 20;
    console.log(a*b);
}
multiply();
```

### Function without parameters - with return types

```
function get():string{
    var s:string = "Hello";
    return s;
}
var text:string = get();
console.log(text);
```

**Arrow Function with parameters - with return types**

```
var sum=(x:number , y: number):number =>{ return x+y; }  
var result:number=sum(10,20);  
console.log(result);
```

**Arrow Function with parameters - without return types**

```
var sum=(x:number,y:number):void => { console.log(x+y); }  
sum(10,20);
```

**Arrow Function without parameters - without return types**

```
var sum =():void => {  
  var x:number=10  
  var y:number=20  
  console.log(x+y)  
}  
sum();
```

**Arrow Function without parameters - with return types**

```
var sum = ():number =>{  
  var x:number=10  
  var y:number=20  
  return x+y;  
}  
var result:number=sum();  
console.log(result);
```

**Define some arrays and iterate using different loops.**

```
//for loop  
var array:number[] = [1,2,3,4,5,6];  
for(var i=0;i<array.length,i++){  
  console.log(array[i]);  
}
```

```
//for in loop  
for(var i in array){  
  console.log(i);  
}
```

```
//for of loop  
for(var i of array){  
  console.log(i);  
}
```

```
//forEach loop  
array.forEach(function (value){  
  console.log(value);  
});
```

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

```
var numbers = new Array();  
var length = numbers.push("nag");  
length = numbers.push("sai");  
length = numbers.push("praveen");  
console.log( numbers );
```

```
var element = numbers.pop();  
console.log("element is : " + element );  
var element = numbers.pop();  
console.log("element is : " + element );
```

**Leap year**

```
function year(a:number):void{  
  if(a%400==0)  
  {  
    console.log("leap year");  
  }  
  else if(a%100 !=0 && a%4==0){  
    console.log("leap year");  
  }  
  else{  
    console.log("not a leap year");  
  }  
  year(3000);  
}
```