



## TypeScript - Getting Started

- Installing TypeScript
- Invoke TypeScript Prompt
- Compiling TypeScript

`npm install typescript`

`tsc`

`tsc fileName.ts`

## Primitive Data Types

- String
- Number
- Boolean
- Date
- Enum
- Array
- Tuple
- Const
- Null
- Any Type (Untyped)
- void

let name: **string** = "Santhosh Vijay";

let score: **number** = 100;

let subscribed: **boolean** = true;

let startDate: **Date** = new Date(2019, 2, 19);

**enum** Color {Blue, Green, Pink};

let skills: **string**[] = ['angular','blockchain'];

let booking: [string, number];

**const** questions: number = 100;

let bookingId: number = **null**;

let data: **any** = "any value";

```
function echo(msg: string): void{  
    console.log(msg);  
}
```

# TYPESCRIPT.JS



## Classes

```
class BookTickets {  
  
    constructor(public booking : MovieBooking) { }  
  
    getBookings() : number {  
        return this.booking.total;  
    }  
}
```

## Abstract Classes

```
abstract class Employee {  
  
    name: string;  
    department: string;  
    experience: number;  
    abstract getEmployeeId(): number;  
}
```

## Inheritance and Interfaces

```
interface Performance {  
  
    getPerformanceRating(): number;  
}  
  
interface Compensation extends Performance {  
  
    calcSalary(basic: number, bonus: number): number;  
}  
  
class Employee {  
  
    name: string;  
    department: string;  
    experience: number;  
}  
  
class Programmer extends Employee implements Compensation {  
    getPerformanceRating(): number {  
        /** write custom logic */  
    }  
    calcSalary(base: number, bonus: number): number {  
        /** write custom logic */  
    }  
}
```

## Scope/Modifiers

```
Public (default) - public name: string;  
  
Protected - protected empId: number  
  
Private - private subscribed: boolean;  
  
Read Only - readonly pi: number = 3.14159  
  
Static - static log(msg: string){  
    console.log(msg)  
}
```