Interfaces

- to create your own user defined datatype

interface.ts

```
export interface User{
        name : string
        age? : number        → optional
        id : number
      email : number

}

let user : User = {name: "John",  id : 1 , email : " "}

interface Employee extends User {

    salary : number
}

let emp : Employee = {name :'John', id : 1 , email : " " , salary : 50000}

export interface login
{
   login() : User;
}
```

class.ts

```
* import {login, User} from "./interface"
        OR
  import * as loginUser from "./interface"

  class Emp implements login
OR                     loginUser.Login
   login() : User {

    return {name: "Harry", id : 2 , email : " "}

}.
              OR
   login() : loginUser.User {

      return {name : "Harry", id : 2 , email : " "}
   }
```

Object destructuring and Array destructuring

Object destructuring - makes it possible to unpack values from arrays into distinct variables.

```
let {name : Fname, email: Login} : User = {name:"Harry", id:1, email:""}

// Fname -> Harry
// Login -> " "
```

Array Destructuring

first create an array

```
let users : User[] = [
    {name : "A", id:1, email:"1@abc.com"},
    {name : "B", id:2, email: " "},
    {     "C"          3           },
]

let [user1, user2] :User =

// user1  -> A
// user2  -> B

let [user1, user2, ...restUsers] : User = [
    {   A   }
    {   B   }
    {   C   }
    {   D   }
]

// user1  - A
   user2  - B
restUsers - C, D ...and so on
```

You dont have to use index again and again to access elements.

When you compile interfaces, all interfaces are removed.
They wont be a part of production bundle.
So,
When to use interface vs classes?

class	interface
- When you have to retain the type of your prod build - prefered during backend	If you can work without creating an object it is best for you.

programming

Interface does not really exist in javascript/typescript

* Decorator - used to modify behavior of class, method or property at runtime
  - used internally by Angular

```
@Component({})  ⟶ Decorator.
class Emp {
    //code
}
```

* Typescript will not stop your code from compiling. It will give errors but code will still compile. So you cannot rely on the error to stop compilation.