#typescript : Types in TypeScript

TypeScript is a superset of JavaScript.
Let's have a gentle introduction of different types available in TypeScript:

1) String
```
let name: string = 'Rizwan';
```

2) Number
```
let age: number = 21;
```

3) Boolean
```
let isEven: boolean = true;
```

4) Array
```
let ages: number[] = [9, 21, 10]
```

5) Any
```
let distance: any[] = ['1000km', 1000]
```

-> any type is compatible with all the other existing types, so we can type any data with it and assign any value to it later.
-> The null and undefined are typed under "any" type.

6) Custom types
```
type Animal = 'cat' | 'dog'
let animal1: Animal = 'cat' //correct
let animal2: Animal = 'bear' //Type '"bear"' is not assignable to type 'Animal'
```

7) Enum
```
enum Brands {
    BMW,
    Ford,
    Audi
```

```
}
Const myCar: Brands = Brands.Ford
```

-> Inspecting myCar returns 1, which is the index held by Ford.

-> Using enum we can also assign custom numeric values as well.

8) Void

```
function test(): void {
    const A = 0;
}
```

-> The void type represents the absence of any type.

9) Type inference

```
let brand = 'BMW'
brand = 123 // shows error -> Type 'number' is not
assignable to type 'string'
```

-> Typing is optional since TypeScript is smart enough to infer the data types of variables and function return values out of context with a certain level of accuracy.
-> When type inference is not possible, it will assign the "any" type to that data.

10)   Partial

```
interface Hero {
   name: string
   power: number
   }
   const hero: Partial<Hero> = {
   name: 'Iron man'
   }
```

-> We use this type when we want to create an object from an interface but include some of its properties, not all of them.

11) Union
```
let age: string | number = '78'
 age = 21
```

-> It is used to mix types

-> It is used when we know what the possible types are

12) Nullable
```
interface Hero {
    powers: number[] | null | undefined
}
```

-> We can use null and undefined, along with the union type to indicate that a property is nullable.

-> link to GitHub Repo containing all post as pdf files:

References:
1) The TypeScript official site
   https://www.typescriptlang.org/
2) The TypeScript official wiki
   https://github.com/microsoft/TypeScript/wiki

follow and add Rizwan Mushtaq Dhudhaal 👷 in your friend list
.
.
.
.
#webdeveloper #webdevelopment #javascriptdeveloper #javascript #typescript