Typescript

Rewind : Javascript Es6 features

- Let and Const
- Arrow Functions
- Template Literals
- Destructuring
- Default Parameters
- Classes and Inheritance

Typescript

- Open source programming language from Microsoft
- Typed superset of javascript
- Compiles down to plain javascript
- Optional static typing and type inference
- Angular and React

Environment set up

- **To check version of node**
- Node –v
- **To Install typescript**
- Npm install –g typescript
- **To check version of typescript**
- tsc –v

Typescript compilation

- **For compilation**
- tsc main.ts
- It creates a main.js file
- **to run**
- node main.js

For Auto compilation

- tsc main –watch

Let and const

- let and const supports block level scopes and cant re declare multiple times.
- Let no need initialize but const need to initialize
- Const is to declare constant variables

Typescript datatypes

- Number
- String
- Boolean
- Any
- Null
- Undefined
- void

Typescript variable declaration

- let name:string="Albin"
- Typescript supports template string means support multiple lines
- Let myStory: string=`hello All
- I m ${name}
- From bangalore
- `;

Null and undefined

let a:null=null;
Here a value always will be null

Let b:undefined = undefined

Null data type

- Can asssign null for other types too
- Let isDone:boolean = null;

Arrays

- Let nums:number[] =[1,2,3];
- Or
- Let nums2: Array<number> =[1,2,3];

Tuple type

- Some times youcan mixed type called tuple type it may contain string and number
- let person: [string,number] = [‘xxx’,123];

Any type

- If you are not sure what type could be you can use any type
- Let anyVal:any=10

Type Inference

- TypeScript infers types of variables when there is no explicit information available

- Example :
- let a=10;
- a="some text"; → Error

Type Inference

- Types are inferred by TypeScript compiler when:
- Variables are initialized
- Default values are set for parameters
- Function return types are determined

Type Inference

- Type Inference works only in initialization
- Example:
- let a;
- a=10;
- a="some text"; → No Error

Multi type

- **Multitype by piping symbol**
- **Let a: string|number**

- Here a can assigned with the type string and number