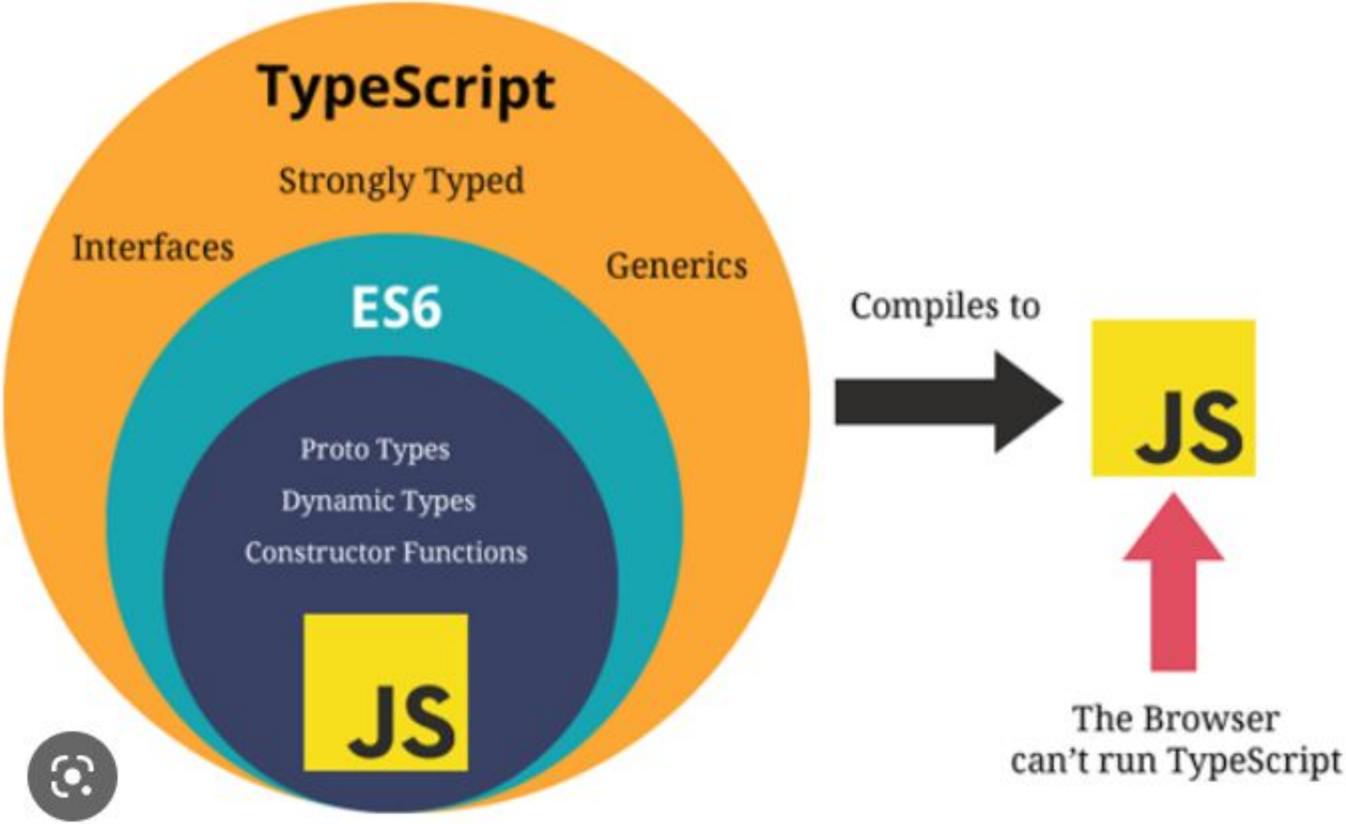
Typescript

Prepared by Faiza Aziz Khan


TypeScript
Strongly Typed
Interfaces
Generics
ES6
Proto Types
Dynamic Types
Constructor Functions
JS
Compiles to
JS
The Browser
can't run TypeScript

What is TypeScript?

1. TypeScript is JavaScript with syntax for types.
2. It is a tool to infer the variable’s type.
3. TS is superset of JS.
4. TypeScript adds additional syntax to JavaScript
5. TypeScript code converts to JavaScript, which runs anywhere JavaScript runs: In a browser, on Node.js
6. Type is data type like numbers, string, null, undefined

What is TypeScript?

6. TS files have .ts extension in the end of filename
7. TS compiler first convert ts to js file then run the code.
8. This extra step is draw back of TS but its still faster than any other tool.

Prerequisites

1. Download node js
2. Install TS
3. Install VScode
4. Check if node js and ts installed
5. Tsc -v and node -v

Install TS

1. Npm i -g typescript
2. In folder tsc –init
3. You have to install **TS** globally to use **tsc** command otherwise you can not.
4. Run tsc to compile ts file to js files
5. Then node filename to run the code in that file

Built In Types

1. String
2. Number
3. boolean
4. Null
5. Undefined
6. bigInt
7. Symbol

Types used for typecasting

1. Any
2. Unknown
3. Never
4. Enum
5. Tuple

Terms in TS

1. Annotate let a: string = “Faiza”
2. Implicit let a= “Faiza”
3. Explicit let a: string = “Faiza”
4. Inferred let a = “Faiza”

Explicit types

```
Let num : number = 1000;

Let name: string=’faiza’;

Let val: boolean=true;

Let response : string | number;
```

Implicit types

Let name = “faiza”

TS inferred from the value that it is string type because value is initialized with string

Let num = 10;

TS inferred from the value that it is number type because value is initialized with number

Let bool = true;

TS inferred from the value that it is boolean type because value is initialized with boolean

Union types

Some time a variable can contain two kinds of type depending on certain conditions.

In that case we can give union type to a variable

```
Let response : string | number;
```

Type alias

Type aliases give a type a new name

```
type type_alias = number | string | boolean;

// Variable is declared of the new type created

let variable: type_alias;

variable = 1;

variable = "im a string value";

variable = true;
```