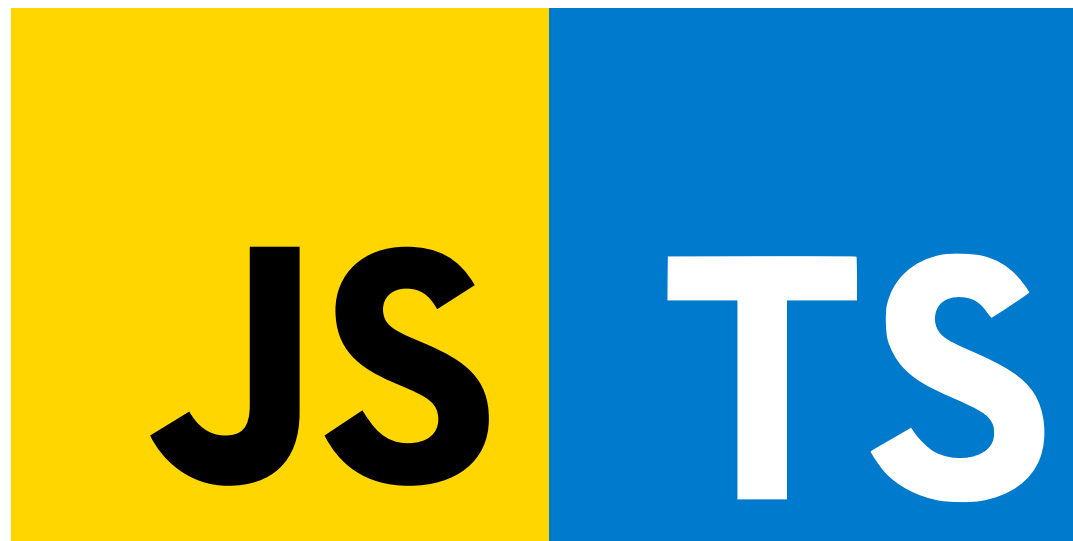




JavaScript

vs.

TypeScript



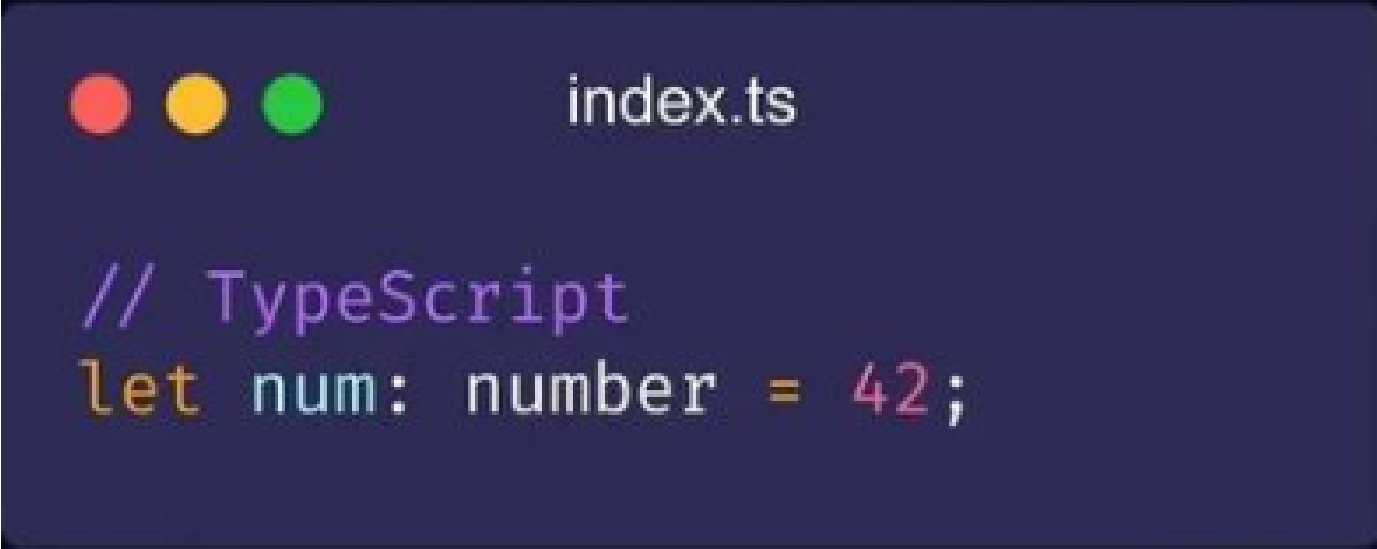
+91-7260058093

www.algotutor.io



1. STATIC TYPING

JavaScript is dynamically typed language.
TypeScript is statically typed language,
which means you must declare variable
types.

A code editor window with a dark blue background. At the top left, there are three colored circles (red, yellow, green) and the filename 'index.ts'. The code is written in a monospaced font with syntax highlighting: a comment '// TypeScript' in purple, and a variable declaration 'let num: number = 42;' where 'let' is orange, 'num:' is light blue, 'number' is white, '=' is white, '42' is pink, and ';' is white.

```
index.ts

// TypeScript
let num: number = 42;
```

2. TYPE INTERFACES

JavaScript lacks type inference. TypeScript infers types based on the assigned values.



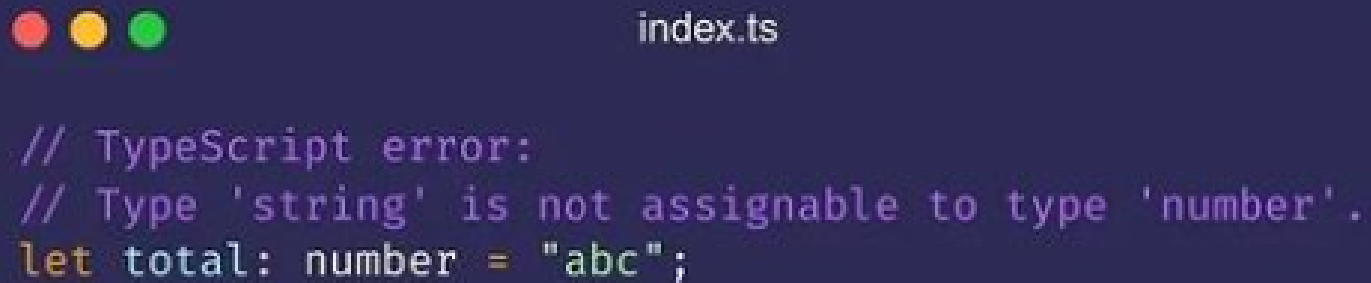
index.ts

```
// TypeScript
```

```
let age = 30; // TypeScript infers age as number
```

3. COMPILE-TIME ERRORS

JavaScript errors are runtime. TypeScript detects errors at compile time due to type checking.



```
index.ts  
  
// TypeScript error:  
// Type 'string' is not assignable to type 'number'.  
let total: number = "abc";
```

4. INTERFACES

JavaScript lacks native support for defining interfaces. TypeScript allows you to define and use interfaces for complex data structures.



```
index.ts

// TypeScript
interface Person {
  name: string;
  age: number;
}

let person: Person = { name: "Alice", age: 25 };
```

5. ENUMS

JavaScript supports no native enums.
TypeScript supports enums for defining set of named constant values.



```
index.ts

// TypeScript
enum Color {
  Red,
  Green,
  Blue,
}

let chosenColor: Color = Color.Green;
```

6. STRICT MODE

"use strict" mode is optional in JavaScript for stricter error handling. TypeScript strict mode is enforced by default.



```
index.ts

// TypeScript will raise errors for unsafe code.
let total = 10;

total.toFixed(2);
// Error: Property 'toFixed' does not exist on type 'number'.
```

6. STRICT MODE

"use strict" mode is optional in JavaScript for stricter error handling. TypeScript strict mode is enforced by default.



```
index.ts

// TypeScript will raise errors for unsafe code.
let total = 10;

total.toFixed(2);
// Error: Property 'toFixed' does not exist on type 'number'.
```