

TS

# JavaScript *VS.* TypeScript



JS TS



**Saad Irfan**  
@DevWithSaad



# 1. Static Typing

JavaScript is dynamically typed language.

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

```
index.ts

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



**Saad Irfan**  
@DevWithSaad



## 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";
```



**Saad Irfan**  
@DevWithSaad



## 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 a 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'.
```



**Saad Irfan**  
@DevWithSaad







**Saad Irfan**

@DevWithSaad

Did You Find it Useful?

Share Your thoughts.

