TypeScript Data Types (With Examples)

1. Primitive Data Types

Primitive types are built-in and immutable:

- number: 10, 3.14, -5
- string: "Hello", 'TS'
- boolean: true, false
- null: null
- undefined: undefined
- bigint: 123456789n
- symbol: Symbol("id")

Examples:

```
let age: number = 25;
let username: string = "Tanveer";
let isLoggedIn: boolean = true;
let salary: bigint = 10000000000n;
let id: symbol = Symbol("userId");
let nothing: null = null;
let notAssigned: undefined = undefined;
```

2. Non-Primitive (Reference) Data Types

Non-primitive types are mutable and stored by reference:

- object: { name: "Ali", age: 30 }
- array: [10, 20, 30]
- tuple: ["Tanveer", 21]
- function: function greet() {}
- class: class Student {}
- enum: enum Direction { Up, Down }

Examples:

let person: { name: string; age: number } = { name: "Ali", age: 30 };

let scores: number[] = [10, 20, 30];

let user: [string, number] = ["Tanveer", 21];

function greet(): void { console.log("Hello, TypeScript"); }

class Student { name: string; constructor(name: string) { this.name = name; } }

enum Direction { Up, Down, Left, Right }

3. Special Data Types

Used in special scenarios:

- any: Accepts any value (no type checking)
- unknown: Like any, but safer
- never: Function that never returns
- void: Function with no return value

Examples:

let random: any = "hello"; random = 10;

let input: unknown = "TS";

if (typeof input === "string") { console.log(input.toUpperCase()); }

function throwError(): never { throw new Error("Error occurred"); }

function logMessage(): void { console.log("This function returns nothing"); }

4. Basic TypeScript Code Summary

// Primitive Types

let age: number = 20;

let name: string = "Ali";

// Non-Primitive

let user: { id: number, name: string } = { id: 1, name: "Tanveer" };

let arr: string[] = ["a", "b", "c"];

let tup: [number, boolean] = [1, true];

// Special

let anything: any = 10;

```
let unkn: unknown = "text";

function test(): void {}

function crash(): never { throw new Error("fail"); }
```

Conclusion

Category | Included Types

---|---

Primitive | number, string, boolean, null, undefined, symbol, bigint

Non-Primitive | object, array, tuple, function, class, enum

Special | any, unknown, never, void