In TypeScript, primitive data types are the most basic types of data that represent single values. They are immutable, meaning that once a primitive data type value is created, it cannot be altered. The primitive data types in TypeScript include:

1. **number**
2. **string**
3. **boolean**
4. **null**
5. **undefined**
6. **bigint**
7. **symbol**

**1. number**

The number type represents both integer and floating-point numbers. TypeScript uses the same number type for all numeric values.

let age: number = 30; let temperature: number = 36.6;

2. string

The string type represents textual data, enclosed in single quotes, double quotes, or backticks for template literals.

let firstName: string = "Alice"; let greeting: string = `Hello, ${firstName}!`;

3. Boolean

The boolean type represents a logical value, which can be either true or false.

let isDone: boolean = true; let hasFinished: boolean = false;

### 4. null

The null type represents the intentional absence of any object value. A variable with null is explicitly set to have no value.

let emptyValue: null = null;

5. undefined

The undefined type indicates that a variable has been declared but not yet assigned a value.

let notInitialized: undefined = undefined;

6. bigint

The bigint type is used for arbitrarily large integers. A bigint is created by appending n to an integer literal or by using the BigInt constructor.

let bigNumber: bigint = 1234567890123456789012345678901234567890n;

let anotherBigNumber: bigint = BigInt("1234567890123456789012345678901234567890");

### 7. symbol

The symbol type is used to create unique identifiers for objects. Symbols are created using the Symbol function.

let uniqueId: symbol = Symbol("id");

8. Unknown

In TypeScript, the unknown type is a top type that represents any value. It is similar to the any type, but safer because TypeScript forces you to perform type checks before you can perform operations on an unknown value. This helps catch potential errors at compile time and encourages safer handling of values that could be of any type.

// number

let distance: number = 42.195; let year: number = 2024;

// string

let color: string = "blue"; let phrase: string = `The year is ${year}.`;

// Boolean

let isActive: boolean = true; let isVerified: boolean = false;

// null

let response: null = null;

// undefined

let result: undefined = undefined;

// bigint

let largeNumber: bigint = 9007199254740991n;

// symbol

let key: symbol = Symbol("key");

console.log(distance, year); // 42.195 2024

console.log(color, phrase); // "blue" "The year is 2024."

console.log(isActive, isVerified); // true false

console.log(response); // null

console.log(result); // undefined

console.log(largeNumber); // 9007199254740991n

console.log(key); // Symbol(key)