

# Estruturas de Dados - TypeScript

## Tipos Primitivos

Para iniciar é essencial entender sobre tipos primitivos para escrever um código mais seguro, legível e eficiente. Os tipos primitivos são os tipos de dados básicos que compõem qualquer aplicação TypeScript. Eles representam valores imutáveis e são usados para armazenar dados fundamentais, como números, textos e valores lógicos.

- number - Representa números inteiros e decimais.

```
let idade: number = 20;
let preco: number = 19.99;
```

- string - Representa cadeias de caracteres.

```
// É possível declarar string com aspas simples(''), duplas("")
// ou template strings(``)

let nome: string = "Felipe";
let professor: string = `O professor é ${nome}`;
```

- boolean - Representa valores lógicos **true** ou **false**.

```
const aprovado: boolean = true;
const reprovado: boolean = false;
```

## Existem alguns tipos em TypeScript que são utilizados em algumas situações especiais

- any - Representa qualquer coisa, ou seja, pode ser qualquer tipo

```
// Deve ser evitado pois tira o sentido de usar TypeScript
// e deixa o código inseguro

let qualquerCoisa: any = 1;
qualquerCoisa = "Sou uma string";
qualquerCoisa = true;
```

- unknown - Representa um valor desconhecido.

```
// semelhante ao any, mas com mais segurança, exigindo
// uma validação antes de utiliza-lo

let valor: unknown;

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

- null e undefined - Representam, respectivamente, ausência de valor e a falta de declaração

```
const vazio: null = null;
const naoDefinido: undefined = undefined;
```

- void e never - Representam, respectivamente, a ausência de retorno de uma função e um valor que nunca pode ser alcançado

```
const imprimirMensagem = (): void => console.log("Hello World!");

// Use never para indicar que uma função não pode
// ter um valor de retorno válido.

function erro(): never {
  throw new Error("Algo deu errado");
}
```

- symbol - Representa um identificador exclusivo de uma propriedade de um objeto

```
const id = Symbol("id");
```

- bigint - Representa um número inteiro grande, além dos limites do tipo number

```
const numeroGrande: bigint = 9007199254740991n;
```