

Aulas Senac

Desenvolvimento Web para Adolescentes

Pablo Garcia

Tipos de dados e variáveis



Introdução aos Tipos de Dados

Definição de Tipos de Dados:

Em programação, um **tipo de dado** define que tipo de valor uma variável pode armazenar e quais operações podem ser realizadas sobre esse valor.



Introdução aos Tipos de Dados

Importância dos Tipos de Dados:

- São essenciais para a correta execução e desempenho de um programa. Eles ajudam o compilador ou o interpretador a gerenciar a memória e executar as operações de maneira eficiente.
- Evitam erros lógicos. Por exemplo, tentar somar um número com uma string pode resultar em comportamento inesperado.

```
7  
8   const n1 = 4;  
9   const n2 = 6;  
10  
11   console.log(n1 + n2);  
12
```

Introdução aos Tipos de Dados

Exemplos em Diferentes Linguagens:

- Em Java, tipos de dados como **int**, **float**, **boolean** são fortemente tipados.
- Já em JavaScript, temos a tipagem dinâmica, ou seja, a variável pode mudar seu tipo de dado conforme o valor atribuído:

javascript

```
let valor = 42; // valor é um número  
valor = "Olá"; // agora valor é uma string
```

Tipos Primitivos em JavaScript

Em JavaScript, os tipos primitivos são os mais básicos e imutáveis. Vamos detalhar cada um:

javascript

```
let idade = 30;  
let altura = 1.75;
```

Tipos Primitivos em JavaScript

- **Number:** Representa tanto números inteiros quanto de ponto flutuante (decimais).
- JavaScript não diferencia entre inteiros e decimais (não existe int ou float, é tudo Number).

javascript

```
let idade = 30;  
let altura = 1.75;
```

Tipos Primitivos em JavaScript

- **String:** Utilizado para armazenar textos.
- Strings podem ser delimitadas por aspas simples, duplas ou crase (para *template strings*).

```
javascript
```

```
let saudacao = "Olá, Mundo!";
```

Tipos Primitivos em JavaScript

- **Boolean:** Representa valores lógicos: **true** ou **false**.

javascript

```
let aprovado = true;  
let reprovado = false;
```


Tipos Primitivos em JavaScript

- **Undefined:** Uma variável que foi declarada, mas ainda não recebeu um valor, possui o valor **undefined**.
- **Null:** Representa intencionalmente a ausência de valor.

javascript

```
let nome;  
console.log(nome); // undefined
```

javascript

```
let contaBancaria = null; // ausência de valor
```

Diferença entre Undefined e Null:

- **Undefined:** Automático quando uma variável é declarada sem valor.
- **Null:** Usado pelo desenvolvedor para definir que a variável não tem valor.

Obrigado

Desenvolvimento Web para Adolescentes

Pablo Garcia

Tipos de dados e variáveis

