Por: Alice Dantas

Variáveis e Tipos em JS

As principais formas de declarar variáveis:

```
var nome = "João";      // Forma antiga
let idade = 25;         // Forma moderna (mais usada)
const PI = 3.14;        // Constante (não pode mudar)
```

Os tipos primitivos mais comuns:

```
let nome = "Ana";        // string
let idade = 30;          // number
let aberto = false;      // boolean
let peso;                // undefined (não tem valor ainda)
let nada = null;         // null (intencionalmente vazio)
```

Testando os tipos com typeof:

```
let nome = "Maria";
console.log(typeof nome);   // "string"

let idade = 20;
console.log(typeof idade);  // "number"

let ativo = true;
console.log(typeof ativo);  // "boolean"
```

Usando variáveis com *Templates Literais*:

- Use crases `` ` `` no lugar de ' ou ".
- Use `${...}` para colocar variáveis ou expressões dentro da string.

```
let nome = "João";
let saudacao = `Olá, ${nome}!`;
console.log(saudacao);   // Olá, João!

let a = 5;
let b = 3;
let resultado = `A soma de ${a} + ${b} é ${a + b}`;
console.log(resultado);  // A soma de 5 + 3 é 8
```

Por: Alice Dantas

Tratamento de Dados em JS

É o processo de **converter** dados de um tipo para outro, **limpar** dados (ex: remover espaços em branco), **formatar** dados para exibição (ex: datas, moedas), etc.

Casting (Conversão de Tipo)

Casting para String

```
String(123);        // "123"
String(true);       // "true"
String(null);       // "null"
String(undefined); // "undefined"
String(NaN);        // "NaN"
```

Casting para Number

```
Number("42");               // 42
Number("3.14");             // 3.14
Number(true);               // 1
Number(false);              // 0
Number(null);               // 0
Number.parseInt(21.5)       //21
Number.parseFloat('21.5')  //21.7
Number("abc");              // NaN
Number(undefined);          // NaN
```

Casting para Boolean

Falsy (valores que se comportam como false em contextos booleanos): `0`, `-0`, `0n`, `false`, `""`, `null`, `undefined`, `NaN`. Qualquer outro valor é considerado truthy, inclusive: `"0"`, `"false"`, `[]`, `{}`, funções, números diferentes de zero etc.

```
Boolean(0);          // false
Boolean("");         // false
Boolean(null);       // false
Boolean(undefined); // false
Boolean(NaN);        // false

Boolean(1);          // true (e qualquer outro valor diferente de 0)
Boolean("abc");      // true (até espaço em branco)
Boolean([]);         // true (mesmo array vazio)
Boolean({});         // true (mesmo objeto vazio)
```

!!valor é o mesmo que Boolean(valor)

- O primeiro **!** **nega** o valor (transforma em booleano e inverte)
- O segundo **!** **inverte de novo**, devolvendo o booleano real

Por: Alice Dantas

Limpeza de dados

```
let nome = "  João da Silva  ";
console.log(nome.trim());       // "João da Silva"
console.log(nome.trimStart());  // "João da Silva  "
console.log(nome.trimEnd());    // "  João da Silva"
```

Formatação de dados

```
let preco = "1.999,90";
let normalizado = preco.replace(".", "").replace(",", ".");
console.log(normalizado);  //”1999.90”

let texto = 'Oi, Pessoal'
console.log(texto.toUpperCase())  // OI, PESSOAL
console.log(texto.toLowerCase())  // oi, pessoal
console.log(texto.length)         // 11
```

Por: Alice Dantas

Operadores em JS

Aritméticos:

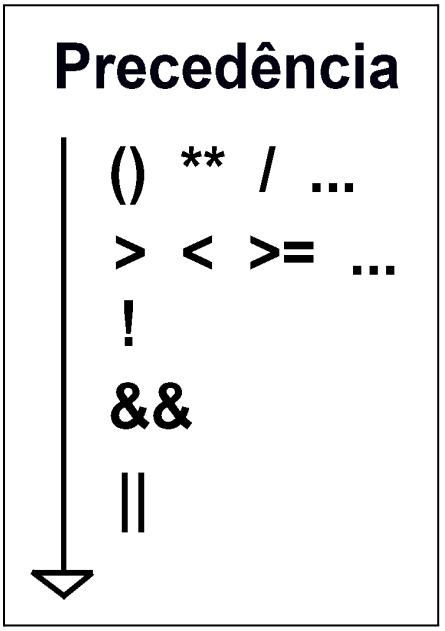
- `+` → Soma; Ex.: `5 + 2`;
- `-` → Subtração; Ex.: `5 - 2`;
- `*` → Multiplicação; Ex.: `5 * 2`;
- `/` → Divisão; Ex.: `10 / 2`;
- `%` → Resto; Ex.: `5 % 2`;
- `**` → Potência; Ex.: `2 ** 3`;

De Atribuição:

- `=` → Atribui; Ex.: `x = 10`
- `+=` → Ex.: `x += 5`; Equivale a: `x = x + 5`
- `-=` → Ex.: `x -= 2`; Equivale a: `x = x - 2`
- `*=` → Ex.: `x *= 3`; Equivale a: `x = x * 3`
- `/=` → Ex.: `x /= 2`; Equivale a: `x = x / 2`

De Comparação (true/false):

Operador	Significado	Exemplo	Resultado
==	Igual (valor)	5 == '5'	true
===	Igual (valor e tipo)	5 === '5'	false
!=	Diferente (valor)	5 != '5'	false
!==	Diferente (valor e tipo)	5 !== '5'	true
>	Maior que	7 > 5	true
<	Menor que	3 < 2	false
>=	Maior ou igual	5 >= 5	true
<=	Menor ou igual	2 <= 1	false



Operadores Lógicos:

`&&` , `||` , `!`

Operadores de Incremento e Decremento:

`++` , `--`

Operador Ternário:

`condição ? valorSeVerdade : valorSeFalso`. Ex.:

```
let idade = 18;
let podeEntrar = idade >= 18 ? "Sim" : "Não";
console.log(podeEntrar); // "Sim"
```