



INSTITUTO FEDERAL  
PIAUÍ  
Campus Parnaíba

# JavaScript

## Variáveis, tipos primitivos e operadores

Prof. Msc Denival A. dos Santos

# Comentários

- Comentário de linha

**// comentário**

- Comentário de múltiplas linhas

**/\***

**Comentário**

**\*/**

# Identificadores

- Podem começar com letra, \$ ou \_
- Não podem começar com números
- É possível usar letras ou números
- É possível usar acentos e símbolos (\*)
- Não podem conter espaços
- Não podem ser palavras reservadas do JavaScript
- Maiúsculas e minúsculas são diferentes, ou seja, JavaScript é case-sensitive.
- Utilize nome sugestivos para as variáveis

# Tipos de dados

- **Number** - JavaScript não diferencia números inteiros de reais
  - Ex.: 1, 5, -12, 9.9, 3.14
  - Tem dois valores internos
    - NaN
    - Infinity
- **String** - cadeia de texto
- **Boolean** - true ou false
- **Undefined**
- **Function** - java permite declarar uma função como variável
- **Object**
  - Array
  - Null

# Tipos de dados

- JavaScript é uma **"loosely typed language"**, o que significa que, diferente de outras linguagens, você não precisa especificar que tipo de dados uma variável irá conter.

```
1  var meuNumero = '500'; // opa, isso continua sendo uma string
2  typeof(meuNumero);
3  meuNumero = 500; // bem melhor – agora isso é um número
4  typeof(meuNumero);
```

# Declarações

- Existem três tipos de declarações em JavaScript

- **Var**

- Declara uma variável, opcionalmente, inicializando-a com um valor.
    - Exemplo: `var a = 10;`

- **Let**

- Declara uma variável local de escopo do bloco, opcionalmente, inicializando-a com um valor.
    - Exemplo: `let a = 'José'`

- **Const**

- Declara uma constante de escopo de bloco, apenas de leitura.
    - Exemplo: `const pi = 3.14`

**Observação:** *Uma variável declarada usando a declaração `var` ou `let` sem especificar o valor inicial tem o valor `undefined`.*

# Operadores aritméticos

Operador	Descrição	Exemplo	Resultado
+	Adição	$x = 5 + 2$	7
-	Subtração	$x = 5 - 2$	3
*	Multiplicação	$x = 5 * 2$	10
/	Divisão	$x = 5 / 2$	2,5
%	Resto da divisão	$x = 5 \% 2$	1
++	Incremento	$x = ++6$ $x = 6++$	7
--	Decremento	$x = --4$ $x = 4--$	3

# Operadores de atribuição

Operador	Exemplo	Sinônimo	Resultado
=	$x = y$	$x = y$	5
+=	$x += y$	$x = x + y$	15
-=	$x -= y$	$x = x - y$	5
*=	$x *= y$	$x = x * y$	30
/=	$x /= y$	$x = x / y$	2
%=	$x \% = y$	$x = x \% y$	0



# Operadores relacionais

OPERADOR	DESCRIÇÃO
<b>==</b>	Igual a
<b>&lt;&gt; ou !=</b>	Diferente de
<b>===</b>	Igual e do mesmo tipo que
<b>!==</b>	Valor e tipos diferentes de
<b>&gt;</b>	Maior que
<b>&lt;</b>	Menor que
<b>&gt;=</b>	Maior ou igual a
<b>&lt;=</b>	Menor ou igual a

# Operadores de comparação

- Em JavaScript os operadores básicos são os mesmos da maioria das linguagens, `<`, `<=`, `>`, `>=`. No entanto, alguns operadores de associados a igualdade tem diferença.
  - `==` - Retorna verdadeiro caso os operandos sejam iguais.
    - Exemplo: `3 == '3' - true`
  - `===` - Retorna verdadeiro caso os operandos sejam iguais e do mesmo tipo.
    - Exemplo: `3 === '3' - false`
  - `!==` - Retorna verdadeiro caso os operandos não sejam iguais e/ou não sejam do mesmo tipo.
    - Exemplo: `3 !== '3' - true`

**Observação:** (`=>`) não é um operador, mas a notação para função de seta. Ou seja, arrow Function que veremos mais tarde no curso.

# Operadores lógicos

OPERADOR	DESCRIÇÃO
! NOT	Negação
&& AND	Conjunção
OR	Disjunção