



Aula 19

**Desenvolver Aplicações
Para Websites**

JavaScript

- Variáveis
 - Globais
 - Podem ser usadas por todas as funções o documento
 - Locais
 - Só podem ser usadas pela função que definiu ela



```
let variavelGlobal;
```

```
function identificador() {  
    let variavelLocal;  
}
```

Convenções de Nomenclatura de Variáveis

- PascalCase
 - A primeira letra de cada palavra é em maiúscula e as outras letras em minúsculas
- camelCase
 - A primeira palavra é toda em minúscula, e as outras palavras tem a primeira letra em maiúscula e as outras letras em minúsculas
- snake_case
 - Todas as letras são minúsculas e separadas por sublinhado

Convenções de Nomenclatura de Variáveis

- kebab-case
 - Todas as letras são minúsculas e separadas por hífen
 - Muito utilizado para nomear arquivos
 - Muito utilizado para nomear propriedades no html e css
 - Não funciona na grande maioria das linguagens de programação, sendo JavaScript uma delas



```
let PascalCase;
```

```
let camelCase;
```

```
let snake_case;
```

JavaScript

- Operadores de Comparação
 - Realizam uma comparação entre dois valores
 - Retorna um valor booleano, de acordo com o resultado da comparação

JavaScript

- Operadores de Comparação
 - ==
 - Valores iguais
 - ===
 - Valores e tipos iguais



//Valores iguais

1 == '1' *//true*

1 == 2 *//false*

//Valores e tipos iguais

1 === 1 *//true*

1 === '1' *//false*

JavaScript

- Operadores de Comparação
 - !=
 - Valores diferentes
 - !==
 - Valores ou tipos diferentes



//Valores diferentes

1 != 1 //false

1 != 2 //true

//Valores e tipos diferentes

1 !== 1 //false

1 !== '1' //true

JavaScript

- Operadores de Comparação

- >

- Maior

- <

- Menor



```
//Maior
```

```
2 > 1 //true
```

```
1 > 2 //false
```

```
1 > 1 //false
```

```
//Menor
```

```
2 < 1 //false
```

```
1 < 2 //true
```

```
1 < 1 //false
```

JavaScript

- Operadores de Comparação

- `>=`

- Maior ou igual

- `<=`

- Menor ou igual



//Maior ou igual

2 >= 1 //true

1 >= 2 //false

1 >= 1 //true

//Menor ou igual

2 <= 1 //false

1 <= 2 //true

1 <= 1 //true

JavaScript

- if
- else
- else if
 - Definem uma estrutura de condição, onde nós definimos um código, que pode ou não ser executado, de acordo com o resultado da condição

Estrutura de Condição IF JavaScript

if (condicao) { ← Início do Código
Código executado se a
condição for verdadeira

Fim do Código → }

else { ← Início do Código
Código executado se a
condição for falsa

Fim do Código → }



```
if (condicao) {  
    //Código a ser executado  
    //caso a condição for verdadeira  
}  
else {  
    //Código a ser executado  
    //caso a condição não for verdadeira  
}
```

JavaScript

- Operadores Lógicos
 - Realizam uma comparação entre dois valores booleanos
 - Retorna um valor booleano, de acordo com o resultado da comparação

JavaScript

- Operadores Lógicos

- && (E)
 - Ambos os valores precisam ser verdadeiros
- || (OU)
 - Apenas um dos valores precisa ser verdadeiro
- ! (Negação)
 - Inverte o valor
 - Se era verdadeiro vira falso
 - Se era falso vira verdadeiro



```
true && true //true
```

```
true && false //false
```

```
false && true //false
```

```
false && false //false
```



true || true //true

true || false //true

false || true //true

false || false //false



```
!true //false
```

```
!false //true
```

JavaScript

- function
 - Parâmetros
 - Dentro dos parênteses da função, é possível definir parâmetros, que são variáveis locais da função, porém que recebem um valor de fora na hora de invocar a função


Estrutura das Funções JavaScript

Palavra-chave Nome da Função Parâmetro

`function` `identificador` `(parametro)` { ← Início da Função

Código da função

} ← Fim da Função

The background features abstract, overlapping shapes in light blue and light orange. A dark blue rectangular box with rounded corners is centered on the page, containing code and a comment. The box has three small colored circles (red, yellow, green) in the top-left corner, resembling a window's title bar.

```
function identificador(parametro1, parametro2) {  
    //Isso é uma função com parâmetros  
}
```



JavaScript

- function
 - return
 - Determina um retorno para a função



```
function identificador() {  
    return retorno;  
}
```

JavaScript

- DOM (Document Object Model)
 - É um modelo de objeto, que permite que programas e scripts acessem e atualizem dinamicamente o conteúdo, a estrutura e o estilo de um documento

JavaScript

- HTML DOM
 - É o DOM do HTML
 - É através dele, que o JavaScript consegue acessar os elementos do documento HTML, e com isso alterar seu conteúdo, estrutura, e até mesmo alterar os estilos aplicados
 - Representado pela palavra-chave document

HTML DOM JavaScript

- getElementById
 - Permite capturar um elemento html, pelo seu id
 - Podemos capturar e até alterar o valor daquele elemento, com a propriedade value



```
document.getElementById('id');
```

```
document.getElementById('id').value = 'Texto';
```


Funções Embutidas JavaScript

- `parseInt`
 - Converte uma string em um número inteiro
- `parseFloat`
 - Converte uma string em um número decimal



```
parseInt('2');
```

```
parseFloat('3.4');
```

JavaScript

- Math
 - É um objeto que permite executar operações matemáticas



JavaScript

- Math
 - round
 - Executa um arredondamento
 - ceil
 - Executa um arredondamento para cima
 - floor
 - Executa um arredondamento para baixo



```
Math.round(2.7); //3
```

```
Math.round(2.2); //2
```

```
Math.ceil(2.7); //3
```

```
Math.ceil(2.2); //3
```

```
Math.floor(2.7); //2
```

```
Math.floor(2.2); //2
```

JavaScript

- Math
 - pow
 - Executa uma exponenciação
 - sqrt
 - Executa uma raiz quadrada



```
Math.pow(3, 2); //9
```

```
Math.sqrt(16); //4
```

JavaScript

- Math
 - min
 - Retorna o menor valor entre vários
 - max
 - Retorna o maior valor entre vários



```
Math.min(3, 6, 4, 10, 2, 8); //2
```

```
Math.max(3, 6, 4, 10, 2, 8); //10
```

JavaScript

- Math
 - random
 - Retorna um número aleatório entre 0 e 1, com exceção do número 1



```
Math.random();
```



```
Math.floor(Math.random() * 100);
```