

INSTITUTO MAUÁ DE TECNOLOGIA



# Desenvolvimento de sites responsivos modernos

Javascript



# Calendário

- 16-08-2018 -> Introdução
- 23-08-2018 -> HTML
- 30-08-2018 -> CSS
- 06-09-2018 -> CSS Effects
- 13-09-2018 -> Javascript
- 20-09-2018 -> Semana de prova (não tem aula)
- 27-09-2018 -> Entrega 1 do projeto
- 04-10-2018 -> Bootstrap
- 11-10-2018 -> Bootstrap
- 18-10-2018 -> jQuery
- 25-10-2018 -> Dúvidas e ajustes no projeto final
- 01-11-2018 -> Dúvidas e ajustes no projeto final
- 08-11-2018 -> Entrega final do projeto



# Javascript

- Linguagem de programação de 'alto nível'
- Fracamente tipada
- Interpretada (Não precisa ser compilada para ser traduzida pela máquina)
- Multi paradigma



# Importando no HTML

- Pode-se importar de forma inline (mas não é recomendado)
- Na maioria dos casos é recomendado criar um arquivo externo e importá-lo antes de fecharmos a tag <body> no html
- Porque não dentro da tag <head>?

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Exemplo Js</title>
  </head>
  <body>
    <h1>Hello World</h1>

    <script>
      alert('Um alert dentro de um javascript INLINE');
    </script>

    <!-- Carregando um arquivo javascript externo -->
    <script src="./script.js"></script>
  </body>
</html>
```



# Principais tipos de dados e literais

```
// String
var nome = 'Dobbin';

// Numérico
var idade = 28;

// Booleano
var isHumano = true;

// JSON (JavaScript Object Notation)
var professor = {
  nome: 'Dobbin',
  idade: 28,
  isHumano: true
};

// Palavra-chave que indica valor nulo
null

// Propriedade superior cujo valor é indefinido
undefined

// Array (Literal)
var vogais = ['a', 'e', 'i', 'o', 'u'];
var primos = [3, 5, 7, 13];
var mix = [1, 'palavra', {a: 123, b: 456}];
```



# Manipulando Objetos Javascript

```
var professor = {  
  nome: 'Dobbin',  
  idade: 28,  
  isHumano: true  
};
```

```
alert(professor.nome); // output 'Dobbin'
```

```
alert(professor['nome']); // output 'Dobbin'
```

```
professor.nome = 'Rodrigo';  
alert(professor.nome); // output 'Rodrigo'
```

```
professor.sobrenome = 'Fellows';  
alert(professor.sobrenome); // output 'Fellows'
```

```
professor['sobrenome'] = 'Fellows';  
alert(professor.sobrenome); // output 'Fellows'
```



# Manipulando Arrays

```
var vogais = ['a', 'e', 'i', 'o', 'u'];  
  
alert(vogais[0]) // output 'a'  
  
vogais.length // 5  
  
vogais.push('z'); // ['a', 'e', 'i', 'o', 'u', 'z'];  
  
vogais.pop(); // ['a', 'e', 'i', 'o', 'u'];
```



# Variáveis

```
// Declara uma variável, opcionalmente, inicializando-a com um valor.
var teste = 'teste';

if (10 >= 5) {
  | var x = 5;
}
console.log(x); // 5

// Declara uma variável local de escopo do bloco, opcionalmente, inicializando-a com um valor.
let teste = 'teste';

if (10 >= 5) {
  | let y = 5;
}
console.log(y); // ReferenceError: y não está definido

// Declara uma constante apenas de leitura.
const teste = 'teste';

// Declara uma variável que recebe uma função como valor
var funcaoA = function () {
  | alert('Isto é uma função');
}

funcaoA();

// Define uma função de nome "funcaoB"
function funcaoB () {
  | alert('Isto também é uma função');
}

funcaoB();
```





# Operadores de comparação

```
const a = 10; // Numérico
const b = '10'; // String
const c = 25; // Numérico
const d = 10; // Numérico

// Igual ==
a == b // true

// Não igual !=
a != d // false

// Estritamente igual ===
a === b // false

// Estritamente não igual !==
a !== b // true

// Maior que >
c > b // true

// Maior que ou igual >=
d >= c // false

// Menor que <
a < d // false

// Menor que ou igual <=
a <= d // true
```



# Condicional IF, ELSE

```
// Exemplo 1
```

```
if (a > b) {  
  alert('A maior que B');  
}
```

```
// Exemplo 2
```

```
if (ticketPrice <= 200) {  
  alert('Irei para o show');  
} else {  
  alert('Não irei para o show');  
}
```

```
// Exemplo 3
```

```
if (media >= 7) {  
  alert('Aluno aprovado');  
} else if (media >= 3) {  
  alert('Aluno vai para recuperacao');  
} else {  
  alert('Aluno reprovado');  
}
```



# Switch

```
switch (expressao) {  
  case 'verde':  
    alert('Cor verde');  
    break;  
  case 'amarela':  
    alert('Cor amarela');  
  case 'azul':  
    alert('Cor azul');  
    break;  
  case 'vermelha':  
    alert('Cor vermelha');  
    break;  
  default:  
    alert('Cor preta');  
    break;  
}
```



# For loop

```
const vogais = ['a', 'e', 'i', 'o', 'u'];
const objeto = {a: 1, b: 2, c: 3};

for (let indice = 0; indice < vogais.length; indice++) {
  alert(indice);           // 0 --> 1 --> 2 --> 3 --> 4
  alert(vogais[indice]); // 'a' --> 'e' --> 'i' --> 'o' --> 'u'
}

for (let indice of vogais) {
  alert(indice); // 'a' --> 'e' --> 'i' --> 'o' --> 'u'
}

for (let indice in objeto) {
  console.log(indice);           // a --> b --> c
  console.log(objeto[indice]); // 1 --> 2 --> 3
}
```



# While loop

```
let x = 0;

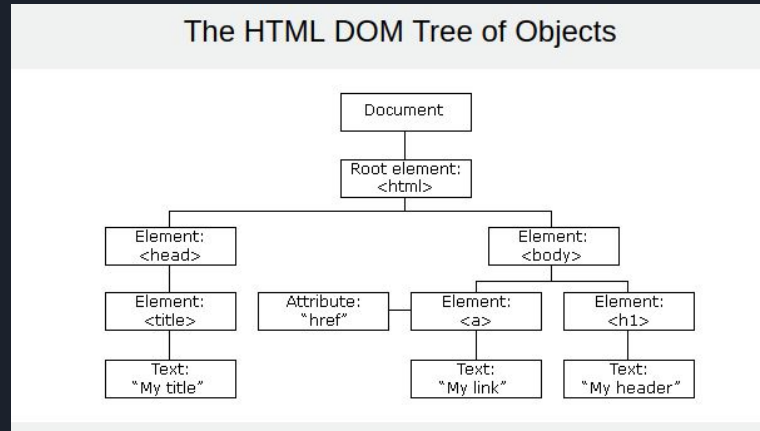
while (x < 10) {
  alert(x);
  x++;
}
```

```
let y = 0;

do {
  alert(y);
  y++;
} while (y < 10)
```

# DOM (Document Object Model)

- Padrão criado pela W3C (World Wide Web Consortium) de definição de acesso a documentos
- Quando uma página web é carregada, o browser cria o DOM da página
- O html DOM define:
  - Elementos html como objetos
  - As propriedades dos elementos html
  - Os métodos de acesso aos elementos html
  - Os eventos dos elementos html
  - Resumindo: o html DOM é um padrão para acessar, alterar, adicionar, deletar elementos html



# DOM (Exemplo utilizando evento)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Exemplo Js</title>
  </head>
  <body>
    <button id="botaoClicavel">Clique Aqui</button>

    <script>
      document.getElementById('botaoClicavel').addEventListener('click', function () {
        alert('O botão foi clicado.');
```



# Referências

- Sintaxe e tipos -> [Mozilla](#)
- Expressões e Operadores -> [Mozilla](#)
- DOM -> [Mozilla](#)





# Exemplos



Perguntas?



# Atividade

- Manipular o DOM do seu projeto utilizando javascript