

```
1
2
3  CAPACITAÇÃO EM 'JAVASCRIPT' {
4
5      [DIA 03]
6
7
8      < PET INFORMÁTICA - UNIVERSIDADE ESTADUAL
9      DE MARINGÁ >
10
11
12  }
13
14
```

Conteúdo do 'Dia 03' {

01 Funções

< O que são e como
utilizá-las />

02 DOM (Document Object Model)

< Entendendo um pouco />

03 Métodos internos

< Praticando />

}

```
1 01 {  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12 }  
13  
14
```

Funções

Funções {

- Um conjunto de instruções que executa uma tarefa ou calcula um valor.
- Se você está repetindo diversas vezes um mesmo bloco de código, muito provavelmente poderá transformá-lo em uma função.

}

Funções {

- Estrutura de uma função:

```
function nomeFuncao(parametro1, parametro2, ...,  
parametroN) {  
    //bloco de instrução da função  
    //pode ou não retornar algo  
}
```

```
}
```

Funções {

- Abaixo um exemplo de uma declaração de um função:

```
function soma(a, b) {  
    return a + b;  
}
```

```
}
```

Funções {

- Para executarmos uma função, precisamos chamá-la:

```
function soma(a, b) {
```

```
    return a + b;
```

```
}
```

```
const resultado = soma(1, 2);
```

```
}
```

Funções {

```
function quadrado(n) {  
    return n * n;  
}  
  
const num = quadrado(5);  
  
}
```


Funções {

```
var quadrado = function(n) {return n * n};  
var x = quadrado(2); //x recebe o valor 4
```

```
}
```

Funções {

- Uma arrow function possui uma sintaxe mais enxuta.

```
let quadrado = n => n * n;  
console.log(quadrado(2)); //4
```

```
}
```

Funções {

```
function busca(nomes, nome) {  
  for (i of nomes) {  
    if (i == nome) return true;  
  }  
  return false;  
}  
  
let nomes = ["Claudio", "Maria", "Marcos", "Daniel"];  
console.log(busca(nomes, "Daniel"));  
}
```

Funções {

```
1
2
3     function somaQuadrados(a, b) {
4         function quadrados(n) {
5             return n * n;
6         }
7         return quadrados(a) + quadrados(b);
8     }
9
10 console.log(somaQuadrados(2, 3)); // 13
11
12
13
14 }
```

Exercícios {

01 - Crie uma função que recebe um nome e mostra no console este nome.

02 - Crie uma função que recebe um nome e sobrenome e retorna o nome completo.

03 - Crie uma função que recebe um vetor de inteiros e retorna a soma destes elementos.

}

```
1 02 {  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14
```

DOM
(Document Object Model)

```
}  
14
```

HTML {

Linguagem de Marcação de Hipertexto

- Linguagem de marcação padrão para criar páginas da Web
- Descreve a estrutura de uma página da Web
- Consiste em uma série de elementos
 - Informam ao navegador como exibir o conteúdo e rotulam partes

}

HTML {

```
<title>, <body>, <header>, <footer>,  
<article>, <section>, <p>, <div>, <span>,  
<img>, <aside>, <audio>, <canvas>,  
<datalist>, <details>, <embed>, <nav>,  
<output>, <progress>, <video>, <ul>, <ol>,  
<li>
```

```
}
```



```
1  HTML {
2
3      <html>
4          <head>
5              <title>Título da página</title>
6          </head>
7          <body>
8              <div>
9                  <h1>Título do texto</h1>
10                 <p>Parágrafo</p>
11             </div>
12         </body>
13     }
14     </html>
```

```
1  HTML {
2
3      <body>
4
5          <div id="texto">
6
7              <h1 id="titulo_texto">Título</h1>
8
9              <p id="paragrafo">texto</p>
10
11          </div>
12      </body>
13
14  }
```

DOM {

- Usado para acessar e manipular o documento HTML.
- O Modelo de Objeto de Documento fornece uma representação estruturada do documento como uma árvore.

}

DOM {

- O DOM define métodos que permitem acesso à árvore, para que eles possam alterar a estrutura, estilo e conteúdo do documento.
- Embora o DOM seja frequentemente acessado usando JavaScript, não é uma parte da linguagem JavaScript. Ele também pode ser acessado por outras linguagens.

}

DOM {

- Para pegarmos o primeiro parágrafo encontrado no documento, podemos utilizar o código abaixo:

```
const paragrafo = document.querySelector('p');
```

```
}
```

DOM {

- O document diz respeito ao próprio documento HTML.
- O método querySelector retorna o primeiro elemento do documento que for igual ao passado como parâmetro para ele.

}

DOM {

- Existem muitos métodos no DOM, alguns deles são:
- `document.querySelector('seletor');`
- `document.getElementById('id');`
- `document.createElement('div');`
- `elemento.setAttribute(nome, valor);`
- `elemento.addEventListener('tipo', função);`

}

```
1  
2 03 {  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12 }  
13  
14
```

Métodos internos

JSON {

- JSON (JavaScript Object Notation) é, como o próprio nome já diz, uma notação de objetos baseada em JavaScript.
- Apesar de ser derivado do JavaScript, também é suportado em diversas outras linguagens de programação (C#, Java, Python).

}

JSON {

- É comumente usado para trocar informações/dados entre clientes e servidores web.
- Para seres humanos é fácil de ler e escrever e para máquinas é fácil de interpretar e gerar.

}

JSON {

- Exemplo:

```
{  
  "nome": "Nay",  
  "idade": 20,  
  "curso": "informatica"  
}
```

```
}
```

JSON {

- Possui dois métodos:

- `JSON.parse()`

Recebe uma string JSON e a transforma em um objeto JavaScript.

- `JSON.stringify()`

Recebe um objeto JavaScript e o transforma em uma string JSON.

}

Fetch API {

- Uma forma de fazer requisições para outras aplicações, interagir com outros sistemas.

```
fetch('url').then(function(resposta) {  
    resposta.json().then(function(resultado) {  
        console.log(resultado);  
    });  
});
```

Fetch API {

- Pegando dados de uma conta do Github:

```
fetch('https://api.github.com/users/yuripiresalves')
  .then(function(resposta) {
    resposta.json().then(function(resultado) {
      console.log(resultado);
    });
  });
}
```

localStorage {

- Permite o acesso ao Storage do navegador.
- O Storage é a interface de Armazenamento da Web que permite que você, por exemplo, adicione, modifique ou exclua itens de dados armazenados.

}

localStorage {

- Se você quiser manipular o armazenamento local para um domínio, você chama o método:

Window.localStorage

}

localStorage {

- Alguns métodos do localStorage:

- localStorage.setItem(chave, valor)

Quando passado um nome de chave e valor, irá adicionar essa chave para o armazenamento, ou atualizar o valor dessa chave, se já existir.

}

localStorage {

- localStorage.getItem(chave)

Quando passado um nome de chave, retornará o valor dessa chave.

- localStorage.removeItem(chave)

Quando passado um nome de chave, irá remover essa chave do armazenamento.

}

Obrigado(a) pela presença 🏆;

{

'Dúvidas?'

pet@din.uem.br

@petinfouem

Comissão organizadora: Diogo, Nayane e Yuri +
Danilo, Gabriel, João Staub e Rold



Instagram

Referências Bibliográficas: MDN - JavaScript



Facebook

CREDITS: This presentation template was
created by **Slidesgo**, including icons by
Flaticon, and infographics & images by **Freepik**

< Please keep this slide for attribution >

}