

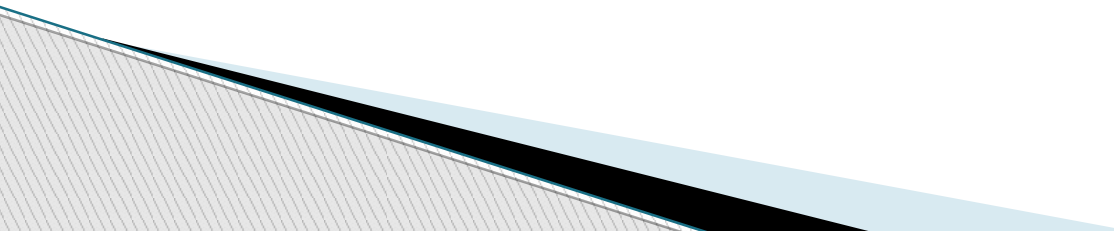
Linguagem de Programação para Web

Objetos e DOM

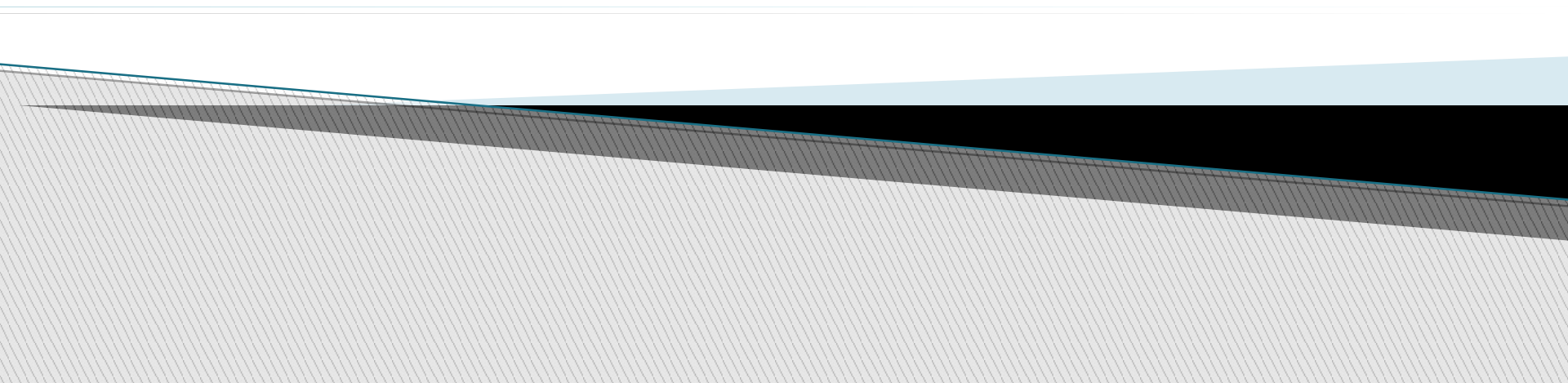
Prof. Tales Bitelo Viegas

A decorative graphic at the bottom of the slide consisting of several overlapping, wavy, horizontal bands. The colors include light blue, dark blue, black, and a grey area with fine diagonal lines.

Valores

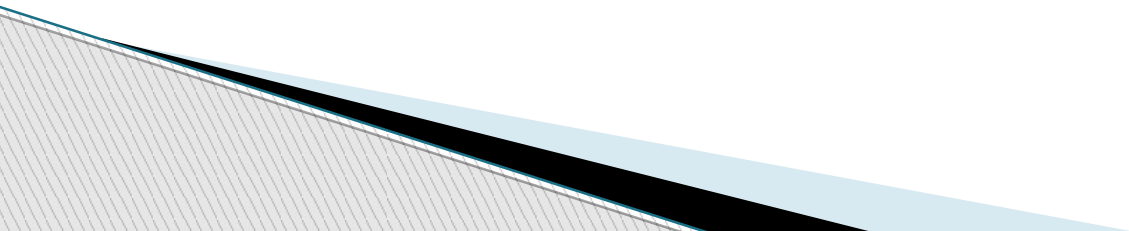
- ▶ Numbers
 - ▶ Strings
 - ▶ Booleans
 - ▶ Objects
 - **null**
 - **undefined**
- 

O resto são Objetos



Objetos


- ▶ Tudo o mais são objetos
- ▶ Podem conter atributos (dados) e métodos
- ▶ Podem ser “herdados” de outros objetos.



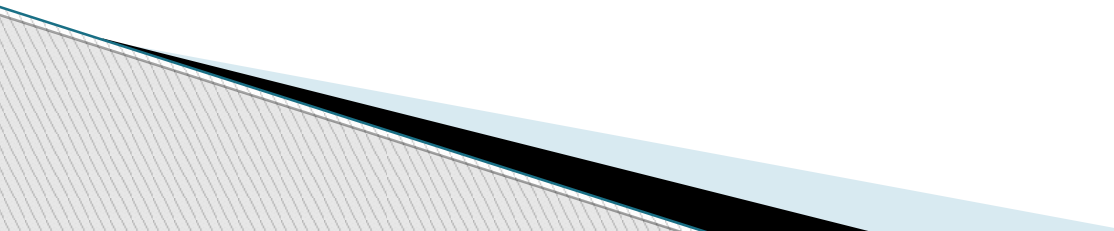
Objetos Dinâmicos

- ▶ Unificação de Object e Hashtable
- **new Object()** produz um container vazio de pares chave/valor
- ▶ Uma chave pode ser qualquer string. Um valor pode ser qualquer tipo, exceto **undefined**
- ▶ Membros (atributos ou funções) podem ser acessados com “.”

Coleções

- ▶ Um objeto é uma coleção não-ordenada de pares chave/valor
 - ▶ Chaves são strings
 - ▶ Valores são qualquer tipo, incluindo outros objetos
 - ▶ Bom para representar registros e árvores
 - ▶ Cada objeto é um “pequeno banco de dados”
- 

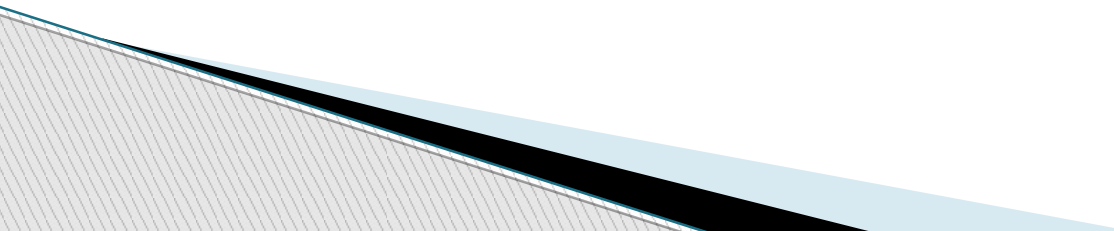
Objetos Literais

- ▶ Objetos literais são delimitados por { }
 - ▶ Chaves podem ser names ou Strings
 - ▶ Valores podem ser expressões
 - : separa nome e valor
 - , separa pares
- 

Exemplo 1 - Objetos Literais

```
var tales = {  
    nome: 'Tales',  
    sobrenome: 'Viegas'  
};
```

```
console.log(tales.nome);  
console.log(tales.sobrenome);
```



Exemplo 2 - Funcao Construtor

```
function Pessoa(nome, sobrenome) {  
    this.nome = nome;  
    this.sobrenome = sobrenome;  
}
```

```
var tales = new Pessoa("tales", "viegas");
```

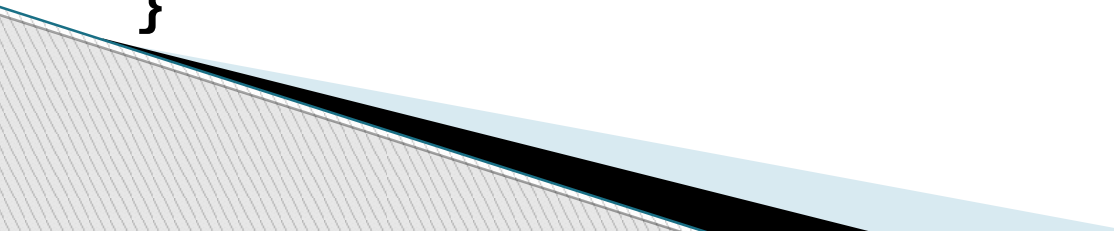


Impressao de Todas as Propriedades

```
// Classe Pai
function Pessoa (nome, sobrenome){
    this.nome = nome;
    this.sobrenome = sobrenome;
}

var tales = new Pessoa("Tales", "Viegas");

for(var propriedade in tales){
    var valor = tales[propriedade];
    console.log(propriedade + " - " + valor);
}
```



Arrays

```
var a = [1,2,3];  
console.log(typeof a);      => object  
console.log(a);             => [1,2,3]  
console.log(a[0]);          => 1  
console.log(a[5]);          => undefined
```

```
a[5] = "string";  
console.log(a);             => [1,2,3,undefined, undefined, "string"]
```

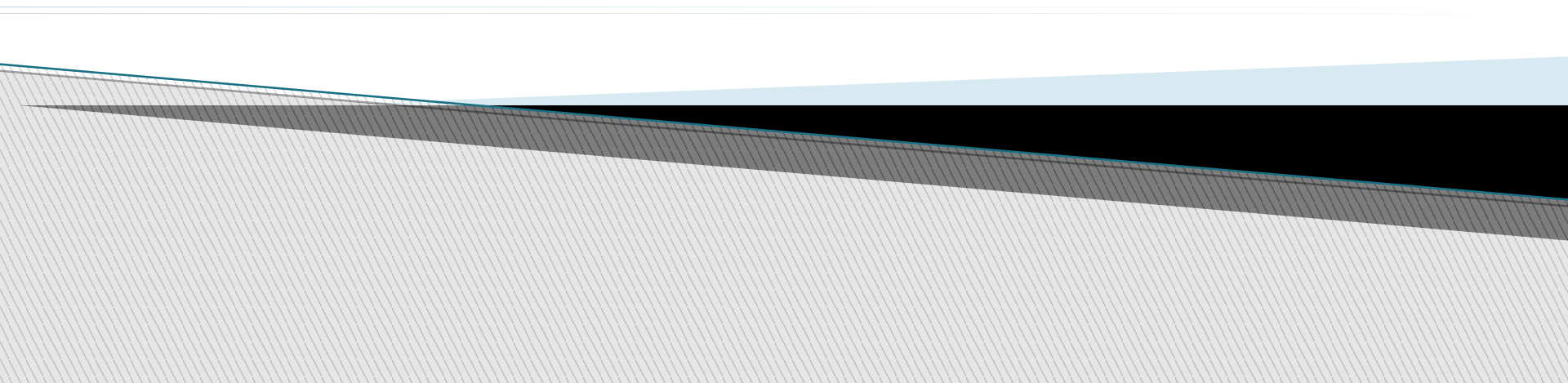
```
delete a[2];  
console.log(a);             => [1,2,undefined,undefined,undefined,"string"]
```

Arrays

```
var a2 = [[1,2,3], ["s1","s2",3]];
console.log(typeof a2);    => object
console.log(a2[0]);        => [1,2,3]
console.log(a2[1]);        => ["s1","s2",3]
```

```
var a3 = "one";
console.log(typeof a3);    => string
console.log(a3[0]);        => "o"
console.log(typeof a3[0]); => string
console.log(a3[1]);        => "n"
```

JavaScript em Páginas Web



Onde colocar

- ▶ Junto com o código do arquivo HTML
 - No HEAD
 - No BODY
- ▶ Em um arquivo externo

Tag `<script>`

- ▶ Dois propósitos
 - Identificar um bloco de script na página
 - Carregar um arquivo de script
- ▶ Detalhe
 - Bloqueia o download de outros componentes da página até que o script seja buscado, compilado e executado.
 - Carregar o script o mais tarde possível (o mais perto da tag `</body>`)

Tag <script> no início

```
<!DOCTYPE html>  
<html lang="pt-br">
```

```
<head>  
<meta charset="UTF-8"/>  
<title>Primeira Página</title>  
</head>
```

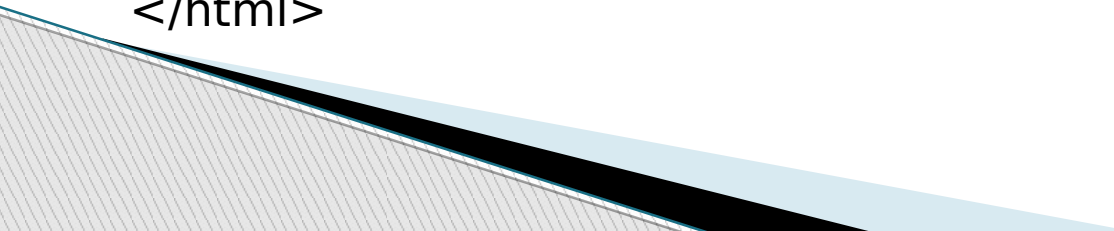
```
<body>  
  <script>  
    alert("Executando Script!");  
  </script>  
  Primeiro conteúdo  
</body>  
</html>
```


Tag `<script>` no final

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
<meta charset="UTF-8"/>
<title>Primeira Página</title>
</head>

<body>
Primeiro conteúdo
<script>
alert("Executando Script!");
</script>
</body>
</html>
```




Utilizando <script src>

```
<!DOCTYPE html>  
<html lang="pt-br">
```

```
<head>  
<meta charset="UTF-8"/>  
<title>Primeira Página</title>  
</head>
```

```
<body>  
<script type="text/javascript" src="03_javascript.js"></script>  
Primeiro conteúdo  
</body>  
</html>
```

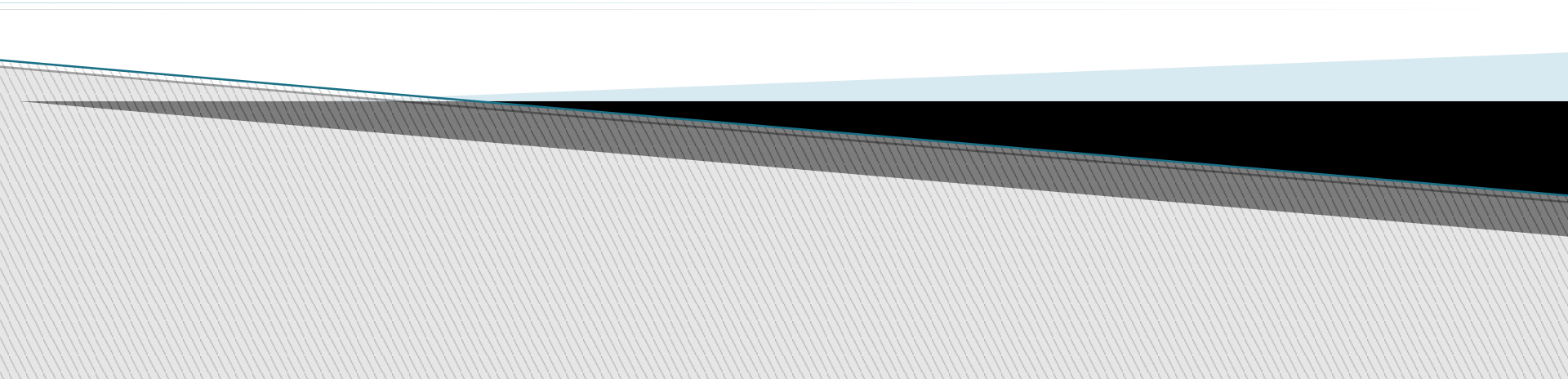
O que o Browser faz?

- ▶ Download da página requisitada
 - ▶ Download dos componentes externos (CSS, JavaScripts, Imagens)
 - ▶ Carga progressiva do conteúdo
 - ▶ Tamanho dos elementos
 - ▶ Posicionamento dos elementos
 - ▶ Interação com o usuário (teclado, mouse)
 - ▶ Gerenciamento dos eventos (teclado, mouse, Ajax, timers)
 - ▶ Cria o DOM, para interagir com o JavaScript
- 

DOM

Document Object

Model



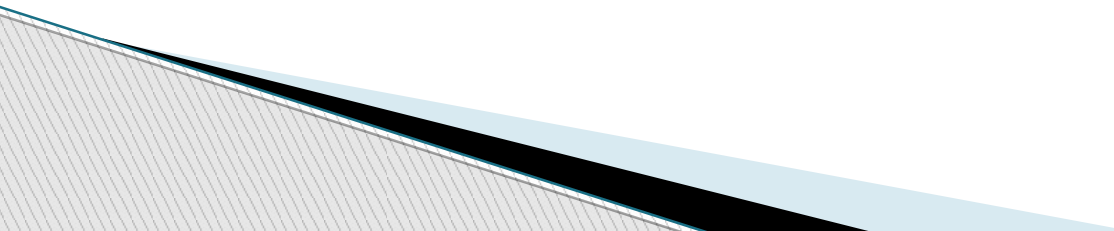
DOM

- ▶ API para validar o HTML e documentos XML bem-formatados
- ▶ Define a estrutura lógica dos documentos e a maneira como um documento é acessado e manipulado

DOM

- ▶ É a interface entre a linguagem JavaScript e os objetos do HTML

Referências

- ▶ Document
 - ▶ Element
 - ▶ Window
 - ▶ Navigator
 - ▶ Screen
 - ▶ History
 - ▶ Location
 - ▶ ...
- 

Manipulação do DOM

- ▶ Basicamente precisamos de 3 métodos
 - `document.getElementById(id)` – retorna o primeiro elemento com o id informado
 - `document.getElementsByTagName(tag)` – retorna um Array com os elementos que possuem a tag informada
 - `Document.getElementsByTagName(name)` – retorna um Array com os elementos que possuem o name informado.

DOM - Encontrando elementos

```
<!DOCTYPE html>
```

```
<html lang="pt-br">
```

```
<head>
```

```
<meta charset="UTF-8"/>
```

```
<title>Mensagens</title>
```

```
<script type="text/javascript" src="05_mensagens.js"></script>
```

```
</head>
```

```
<body>
```

```
<label for="mensagem">Digite uma Mensagem</label>
```

```
<input type="text" name="mensagem" id="mensagem" />
```

```
<input type="button" value="Enviar" onclick="adiciona()" />
```

```
<div id="listaMensagens"></div>
```

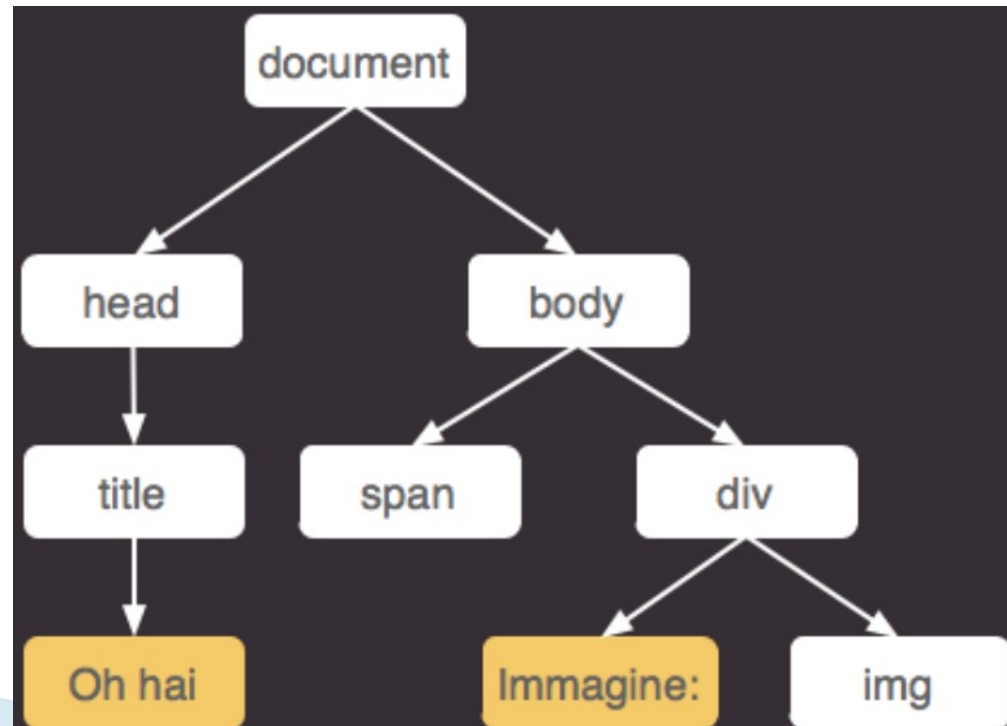
```
</body>
```

```
</html>
```



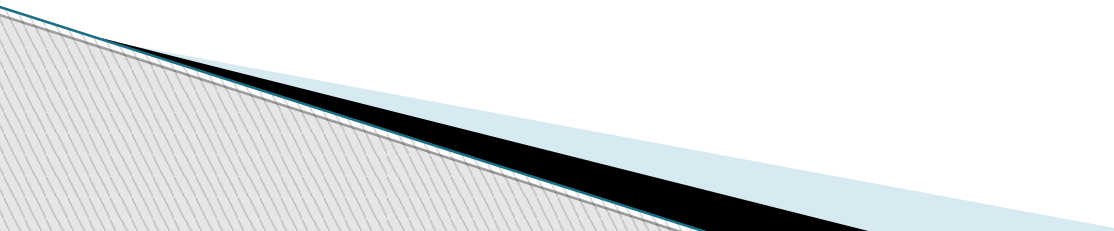
DOM

```
<html>  
  <head>  
    <title>Oh hay</title>  
  </head>  
  <body>  
    <span></span>  
    <div>Imagine:  
    <img src=""/>  
    </div>  
  </body>  
</html>
```



DOM - Encontrando Elementos

```
var adiciona = function(){  
    var objMensagem =  
document.getElementById('mensagem');  
    var mensagem = objMensagem.value;  
    var divResult =  
document.getElementById('listaMensagens');  
  
    divResult.innerHTML += "<p>" + mensagem + "</p>";  
}
```



Exercício 1

- ▶ Crie uma página Web que simule uma calculadora com 4 elementos.
- ▶ Ela deverá possuir:
 - um botão para cada número (0 a 9)
 - um botão para cada operação (+, -, *, /)
 - um botão para o ponto
 - um botão para limpar
 - um botão de igual para o resultado

Exercício 2

- ▶ Crie uma página Web onde o usuário possa escolher uma imagem como fundo da página
- ▶ Você deve mostrar 4 imagens e, quando o usuário clicar em uma delas, o fundo da página é alterado via Javascript