

# FIAP

Abstract flowing lines in shades of pink and blue, resembling smoke or liquid, set against a dark background.

## RESPONSIVE WEB DESIGN

10  
semestre  
2018

[www.fiap.com.br](http://www.fiap.com.br)

RENATO BORTOLIN  
[renatoboschini@gmail.com](mailto:renatoboschini@gmail.com)

# Agenda

1. Complementando o FOR
2. Vamos exercitar o que aprendemos
3. Funções

# ■ Javascript

## Bloco de repetição – FOR IN

Constantemente utilizamos o bloco **for** para iterar sobre atributos de uma coleção, por isso foi criado o bloco **for in**, iterando sobre um array.

```
10 ▾ <script>
11     var cidades = ["São Paulo", "Rio de Janeiro", "Brasilia", "Minhas
12         Gerais"];
13     for (cidade in cidades) {
14         console.log(cidade);
15     };
16 </script>
```

No exemplo acima, recebemos o índices dos elementos do array como resposta.

# ■ Javascript

## Bloco de repetição – **FOR IN**

Portanto para obter o valor de cada índice precisamos implementar isso no bloco:

```
10     <script>
11         var cidades = ["São Paulo", "Rio de Janeiro", "Brasilia", "Minhas
12             Gerais"];
13         for (cidade in cidades) {
14             console.log(cidades[cidade]);
15         };
16     </script>
```

# ■ Javascript

Vamos exercitar o que aprendemos

Vamos fazer um script onde irá receber uma sigla de um determinado estado, **no mínimo 4**.

Este script precisa identificar qual é a cidade. Ao identificar a cidade, mostrar os times de futebol que pertence aquela cidade. **no mínimo 4**.



# ■ Javascript

## Funções

A execução do JavaScript, quando interpretada pelo navegador é imediata. Assim como o HTML interpretado linha a linha, o JavaScript é executado assim que uma tag `<script>` é encontrada em nosso documento.

No entanto, muitas vezes desejamos definir um Comportamento para ser executado em outro momento.

Para isso existem as funções, com a seguinte estrutura básica .

## Funções

```
<script>  
// estrutura de uma função  
    Function nomeDaFuncao(parametro)  
    {  
        //Corpo da Função  
    }  
</script>
```



# ■ Javascript

## Funções

Com base na estrutura anterior, vamos fazer uma que exibe uma mensagem com alert().

```
10      <script>
11          function exibeMensagem(){
12              alert("Fiap Tecnologia");
13          }
14
15          //chamando a mensagem declarada
16          exibeMensagem();
17      </script>
```



# ■ Javascript

## Funções

No exemplo anterior, criamos uma função simples, mas pode ser interessante recebermos a mensagem com parâmetro. Uma função pode receber um ou mais parâmetros.

```
10▼  <script>
11
12
13      function exibeMensagem(mensagem){
14          console.log(mensagem);
15      }
16
17      //chamando a mensagem declarada
18      exibeMensagem("Técnicas e Layouts de Web Sites");
19  </script>
```

# ■ Javascript

## Funções

Geralmente as funções fazem mais do que exibir mensagens. A função exibe a soma de dois números passando como parâmetro:

```
10      <script>
11
12
13      function somaDoisNumeros(valorA, valorB){
14          console.log(valorA + valorB);
15      }
16
17      somaDoisNumeros(43,62);
18      </script>
```

# ■ Javascript

## Funções – com retorno

No exemplo anterior, a função somaDoisNumeros(),  
exibe o resultado da soma, mas muitas vezes,  
precisamos do resultado.

```
10      <script>
11          var numero = 80;
12
13          function somaDoisNumeros(valorA, valorB){
14              return (valorA + valorB);
15          }
16
17          var resultado = somaDoisNumeros(52,14);
18
19          alert(numero - resultado);
20      </script>
```

## Funções – com variáveis

```
10      <script>
11          function exibeMensagem (parametro) {
12              alert(parametro);
13          }
14
15          var funcaoComVariavel = exibeMensagem;
16          funcaoComVariavel("Atenção") ;
17
18      </script>
```

## Funções – com função

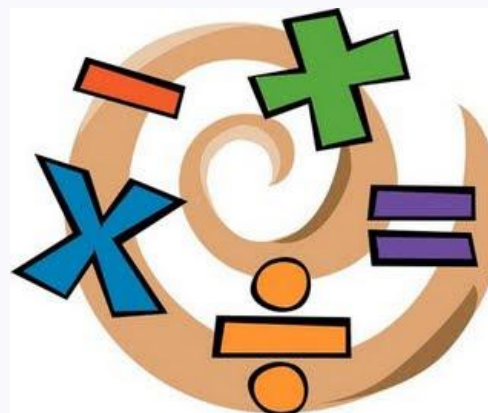
```
10      <script>
11          function exibeAlerta1 () {
12              alert("Atenção")
13          }
14
15          function exibeAlerta2 () {
16              alert("Cuidado")
17          }
18
19          function exibeAlertas(funcao1, funcao2){
20              exibeAlerta2();
21              exibeAlerta1();
22          }
23
24          exibeAlertas();
25      </script>
```

# ■ Javascript

Vamos exercitar o que aprendemos

Vamos criar uma função que receba dois valores. Após o recebimento, a função precisa mostrar o resultado dos seguintes operadores:

- Soma
- Subtração
- Multiplicação
- Divisão



# ■ Javascript

## Eventos



Uma das principais interfaces que os navegadores nos fornecem para que possamos utilizar em conjunto com o nosso script é a interface de eventos.

Com ela é possível que nosso código seja executada a partir de uma ação do usuário em relação do HTML, por exemplo, clicar em um botão, em um link ou simplesmente passar o cursor do mouse sobre uma determinada área.





## Eventos



Todos os elementos do documento respondem à uma série de eventos diferentes. Segue uma lista dos principais eventos suportados nos principais navegadores



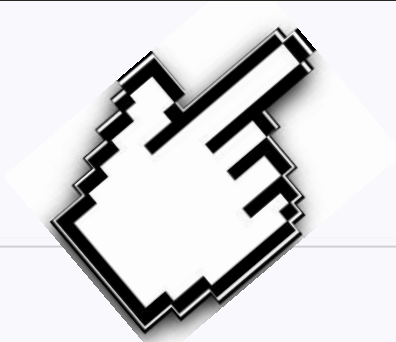
# ■ Javascript

## Eventos



Um evento simples de um link chamando um script

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5 </head>
6 <body>
7   <a href="javascript:alert('Voce clicou em um link!')">
8     <h2>Link com ação JavaScript</h2>
9   </a>
10 </body>
11 </html>
```



# ■ Javascript

## Eventos



## Adicionando um evento click

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5 </head>
6 <body>
7   <a href="#" onclick="alert('Você clicou em um novo link de evento')">
8     <h2>Link com ação JavaScript</h2>
9   </a>
10 </body>
11 </html>
```



# ■ Javascript

## Eventos



**blur:** quando o usuário “sai” de um campo de formulário



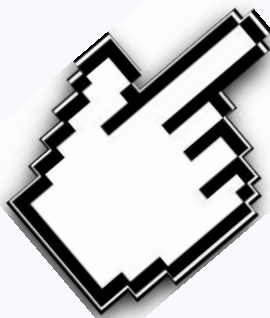
```
1  <!doctype html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5  </head>
6  <body>
7
8      <form>
9          Data de Aniversário
10         <input type="text" value="" onblur="valida();">
11     </form>
12
13     <script>
14         function valida(){
15             alert('Saiu do campos!');
16         }
17
18     </script>
19
20 </body>
21 </html>
```

# ■ Javascript

## Eventos



**change:** quando o usuário muda o conteúdo de um campo



```
<form name="form1">
  Cidade:
  <select onchange="valida();" id="campo">
    <option value=""></option>
    <option value="São Paulo">São Paulo</option>
    <option value="Rio de Janeiro">Rio de Janeiro</option>
    <option value="Bahia">Bahia</option>
  </select>
</form>

<script>
  function valida(){
    alert(document.form1.campo.value);
  }
</script>
```

# ■ Javascript

## Eventos



**click:** quando o usuário clica sobre um elemento

```
8      <form name="form1">
9          Cidade: <a href="" onclick="valida(); valida2();">
10              São Paulo
11          </a>
12
13          <div style="background:red; width:80px; height:80px; cursor:help;"
14              onclick="valida();"
15          </div>
16      </form>
17
18      <script>
19          function valida(){alert("ação do click");}
20
21          function valida2(){alert("ação do click 222");}
22      </script>
```

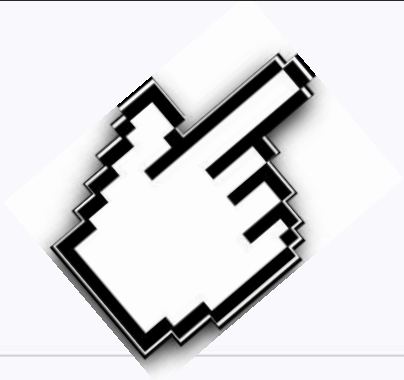
# ■ Javascript

## Eventos



**dblclick:** quando um usuário realiza um duplo click

```
9      <div style="background:blue; width:80px; height:80px; cursor:help;"  
      ondblclick="valida();">  
10     </div>  
11  
12  
13     <script>  
14         function valida(){alert("ação do duplo click");}  
15     </script>
```



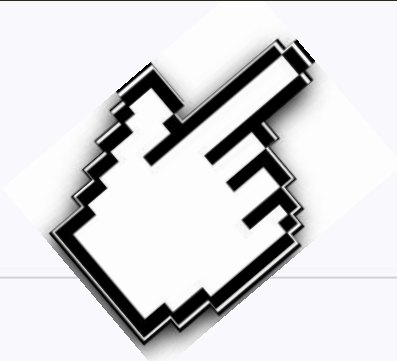


## Eventos



**error:** quando ocorre erro no carregamento de uma página ou numa imagem.

```
9      
10
11      
13
14      <script>
15          function erroImagem(){alert("imagem não encontrada!");}
16      </script>
```



# Javascript

## Eventos



**focus:** quando um elemento é selecionado para interação

```
9      <form>
10      Nome: <input type="text" name="seu_nome" onfocus="entra(this);"
11            onblur="sai(this);">
12      <br>
13      E-mail: <input type="text" name="seu_email" onfocus="entra(this);"
14             onblur="sai(this);">
15      </form>
16
17
18      <script>
19          function entra(campo)
20          {
21              campo.style.background="yellow";
22          }
23
24          function sai(campo)
25          {
26              campo.style.background="#FFFFFF";
27          }
28      </script>
```

# ■ Javascript

## Eventos



**keypress:** quando uma tecla do teclado é pressionada

```
9      <form>
10      Nome: <input type="text" name="seu_nome" onkeypress="valida();">
11      </form>
12
13
14      <script>
15          function valida()
16          {
17              alert('você digitou algo!');
18          };
19      </script>
```



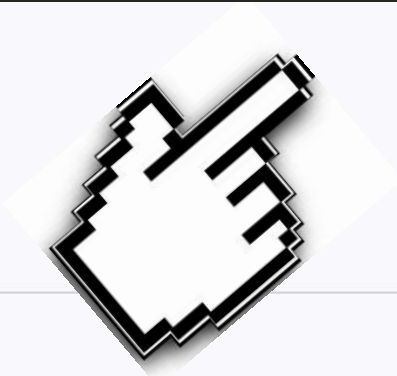
# ■ Javascript

## Eventos



**keyup:** quando a tecla do teclado é solta depois de pressionar

```
9      <form>
10      Nome: <input type="text" name="seu_nome" onkeyup="valida();">
11      </form>
12
13
14      <script>
15          function valida()
16          {
17              alert('você digitou algo!');
18          };
19      </script>
```



# ■ Javascript

## Eventos



**load:** quando a página ou a imagem é carregada

```
6 <body onload="carregaTexto();">
7
8
9 
11
12 <script>
13     function carregaTexto()
14     {
15         alert('Boa noite galera!');
16     };
17
18     function ImagemCarregada(){
19         alert('Imagem carregada!');
20     }
21 </script>
22
23 </body>
```

# ■ Javascript

## Eventos



**mousemove:** quando o mouse está em movimento

```
7 <body>
8     
10
11     <script>
12         function valida(){
13             alert('Botão em movimento!')
14         }
15     </script>
16 </body>
```



# ■ Javascript

## Eventos



**mousedown:** quando o mouse é movimentado em cima do elemento

```
7 <body>
8     <form name="formulario">
9         <input type="button" value="Enviar" onmousedown="valida()">
10
11         
14     </form>
15     <script>
16         function valida(){
17             alert('Botão pressionado!')
18         }
19     </script>
20 </body>
```



# Javascript

## Eventos



**mouseout:** quando um botão do mouse é pressionado

```
7 <body>
8     <form name="formulario">
9         <input type="button" value="Enviar" onmousedown="valida()">
10
11         
14     </form>
15
16     <script>
17         function valida(){
18             alert('Botão pressionado!')
19         }
20     </script>
21 </body>
```

# Javascript

Vamos exercitar



**1º valor**

**2º valor**

**Resultado**

$+$

$-$

$\times$

$\div$



# FIAP



Copyright © **2018** Prof. Renato Bortolin Boschini

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).