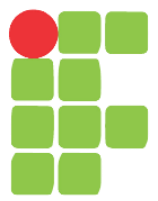


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SERGIPE

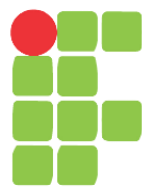
Construção de Sites 2

Prof. Christiano Lima Santos



Conteúdo do Curso

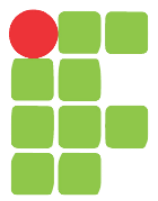
- ▶ JavaScript
- ▶ Framework Bootstrap



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SERGIPE

Introdução a JavaScript

Parte 01



Sumário

- ▶ O que é JavaScript?
- ▶ História de JavaScript e outras linguagens web
- ▶ Editores para JavaScript
- ▶ Como usar JavaScript em uma página
- ▶ Saiba mais



O que é JavaScript?

- ▶ **Linguagem de programação** interpretada client-side com tipagem dinâmica (linguagem de script);
 - ▶ Linguagem de programação - permite a criação de rotinas (conjuntos de instruções) com finalidades específicas. Ex: validar entradas em um formulário, alterar textos, tags ou propriedades CSS de uma página etc.



O que é JavaScript?

- ▶ Linguagem de programação **interpretada** client-side com tipagem dinâmica (linguagem de script);
 - ▶ Interpretada - não é compilada, isto é, o código escrito pelo desenvolvedor é lido e executado pelo interpretador (neste caso, o navegador).



O que é JavaScript?

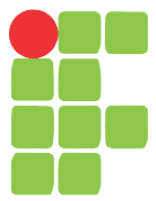
- ▶ Linguagem de programação interpretada **client-side** com tipagem dinâmica (linguagem de script);
 - ▶ Client-side - é executada no computador do cliente (solicitante), não no servidor¹.

¹ Hoje, JS já começa a ser usada no ambiente servidor por meio de node.js



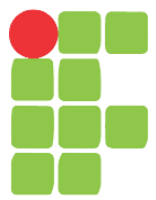
O que é JavaScript?

- ▶ Linguagem de programação interpretada client-side com **tipagem dinâmica** (linguagem de script);
 - ▶ Tipagem dinâmica - variáveis podem receber dados de qualquer tipo.



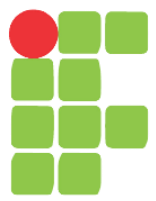
História de JavaScript e outras linguagens web

- ▶ Antes do surgimento das linguagens de script para web, as páginas eram geralmente estáticas e ofereciam poucas formas de interação, limitando-se a hyperlinks e formulários.



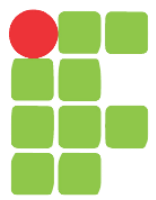
História de JavaScript e outras linguagens web

- ▶ Netscape (líder em navegadores) lança em setembro de 1995 o navegador Netscape 2.0 com suporte a uma nova linguagem, LiveScript;
- ▶ Em dezembro de 1995, em anúncio conjunto com a Sun Microsystems, muda o nome para JavaScript e adiciona suporte à tecnologia Java em seu navegador (applets);
 - ▶ Estratégia de marketing!



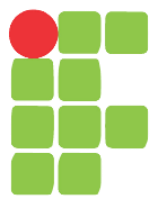
História de JavaScript e outras linguagens web

- ▶ Devido à sua popularidade, Microsoft desenvolveu duas linguagens de script para web: VBScript (deriva do Visual Basic) e JScript (similar a JavaScript, mas não 100% compatível), esta última suportada pelo Internet Explorer 3.0 a partir de agosto de 1996.



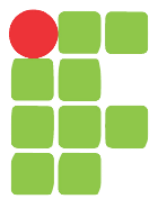
História de JavaScript e outras linguagens web

- ▶ Novembro de 1996 - Netscape é submetida e aprovada como padrão industrial, resultando na versão padronizada ECMAScript / ECMA-262;
- ▶ JScript (Microsoft) não se encontra em conformidade com a especificação ECMA e VBScript é suportada apenas por navegadores Internet Explorer. Além disso, pragas escritas em VBScript prejudicam ainda mais a imagem desta linguagem.



História de JavaScript e outras linguagens web

- ▶ Hoje, JavaScript é um padrão em programação client-side, tendo seu uso aliado a outras tecnologias:
 - ▶ jQuery
 - ▶ JSON
 - ▶ Ajax



Compatibilidade nos browsers

- ▶ Por ser uma linguagem que sofre poucas alterações (atualmente na versão 1.8.5¹, de março de 2011), maioria dos novos browsers suporta (quase) todas suas características.

¹ <https://pt.wikipedia.org/wiki/JavaScript>



Editores para JavaScript

- ▶ Pode-se escrever código JavaScript em qualquer editor de texto...
 - ▶ Bloco de Notas, Notepad++, Brackets etc.
 - ▶ Lembre-se apenas de salvar com a extensão “.js”!
- ▶ ...e ver o resultado produzido em qualquer navegador...
 - ▶ Google Chrome, Mozilla Firefox, Internet Explorer, Safari etc.



Editores para JavaScript

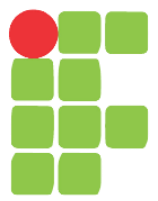
- ▶ Entretanto, há editores JavaScript que oferecem facilidades e recursos para o trabalho do desenvolvedor Web:
 - ▶ Adobe Dreamweaver (pago)
<http://www.adobe.com/products/dreamweaver.html>
 - ▶ WebStorm (pago)
<https://www.jetbrains.com/webstorm/>
 - ▶ Eclipse (gratuito)
<http://www.eclipse.org/downloads/>



Como usar JavaScript em uma página

Forma #1 - aplicando diretamente à tag HTML

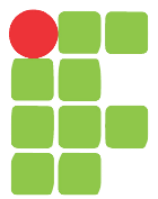
```
<a href="#" onclick="alert('Olá!')">Teste</a>
```



Como usar JavaScript em uma página

Forma #2 - dentro de uma `<script>` na página

```
<script>  
  alert('Olá!');  
</script>
```



Como usar JavaScript em uma página

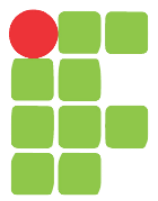
Forma #3 - em um arquivo .js

exemplo-1.3.html

```
<script src="exemplo-1.3.js"></script>
```

exemplo-1.3.js

```
alert('Olá');
```



Depurando scripts



- ▶ Os navegadores implementam ferramentas que podem ser usadas para identificar erros em scripts bem como para testar funcionalidades;
- ▶ No Google Chrome e Mozilla Firefox, por exemplo, há o “Console”, disponível em “Ferramentas do Desenvolvedor”.



Observações

- ▶ JavaScript aceita comentários em linha `//` e em bloco `/* */`;
- ▶ Se cada instrução ocupar uma linha, uso do ponto-e-vírgula `;` no final de cada não é obrigatório, mas é recomendado;
- ▶ JS é uma linguagem case-sensitive, isto é, faz distinção entre letras maiúsculas e minúsculas (a variável “casa” é diferente da variável “Casa”).



Saiba mais

- ▶ **Guia JavaScript (Mozilla)**

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide>

- ▶ **Apostila Desenvolvimento Web com HTML, CSS e JavaScript (Caelum)**

<https://www.caelum.com.br/apostila-html-css-javascript>

- ▶ **JavaScript Tutorial (W3Schools)**

<https://www.w3schools.com/js/default.asp>



Document Object Model e Eventos

Parte 02



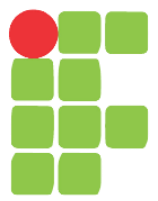
Sumário

- ▶ Modelo DOM
- ▶ Propriedades e Métodos
- ▶ Objetos window, document, location, history e navigator



Modelo DOM

- ▶ Documento Object Model (DOM) refere-se a um modelo de objetos de um documento **independente de plataforma**, que permite a scripts acessar e atualizar conteúdos;
- ▶ A manipulação é feita a partir de propriedades e métodos de objetos ligados ao documento.



Principais objetos DOM

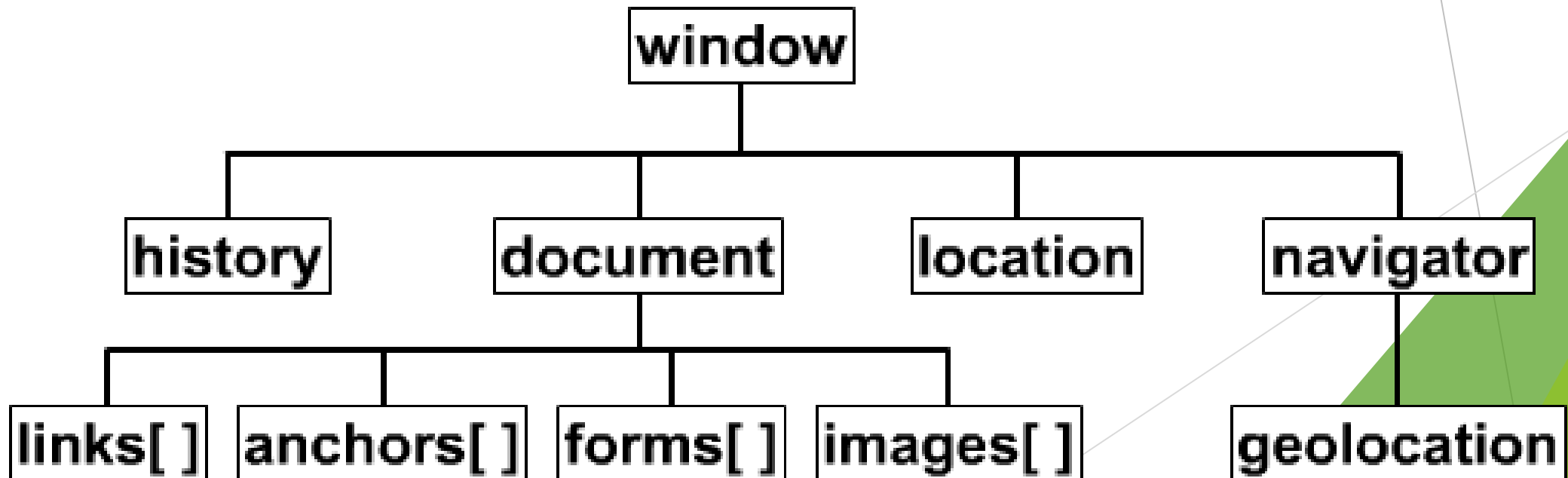
window: refere-se à janela, o objeto de nível mais alto;

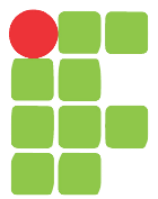
document: refere-se ao documento em si;

location: corresponde à URL da página atual;

history: lista de URLs visitadas pelo usuário até o momento;

navigator: corresponde ao próprio browser.





Objeto window¹

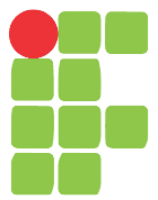


2.1

Propriedade	Descrição
name	Nome da janela
parent	Janela “pai”
closed	Indica se a janela foi fechada ou não
frames	Lista de iframes
frameElement	Iframe que contém a janela

Método	Descrição
alert(...)	Abre caixa de “alerta” com uma mensagem
confirm(...)	Abre caixa de “confirmação” com botões OK e Cancelar
prompt(...)	Abre caixa solicitando entrada de dados do usuário
open(...)	Abre uma nova janela
close()	Fecha a janela
focus()	A janela recebe o foco
blur()	A janela perde o foco
resizeTo(L, A)	Redimensiona a janela
moveTo(X, Y)	Move a janela

¹ https://www.w3schools.com/jsref/obj_window.asp



Objeto document¹

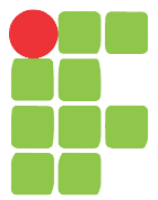
Propriedade	Descrição
bgColor	Cor de fundo
fgColor	Cor do texto
linkColor	Cor dos links
vlinkColor	Cor dos links já visitados
alinkColor	Cor do link ativo (clicado)
title	Título do documento
cookie ²	Dados do cookie do documento atual

Método	Descrição
write(...)	Escreve os valores especificados no documento
writeln(...)	Escreve os valores especificados e, ao final, adiciona uma quebra de linha
getElementById(...) ³	Retorna o elemento cujo <i>id</i> foi especificado
getElementByName(...)	Retorna o elemento cujo <i>name</i> foi especificado

¹ https://www.w3schools.com/jsref/dom_obj_document.asp

² https://www.w3schools.com/js/js_cookies.asp

³ Também é possível pegar um HTML Element apenas usando seu id



Objeto location¹

2.3



Propriedade	Descrição
href	URL da página
protocol	Protocolo (http, https etc.)
hostname	Hostname (domínio)
port	Porta
search	Querystring (?atributo=valor)
hash	Hash/Âncora (#hash)

Método	Descrição
assign(...)	Carrega nova página
reload()	Recarrega página atual
replace(...)	Carrega nova página, porém substituindo a atual no histórico

¹ https://www.w3schools.com/jsref/obj_location.asp

Objeto history¹

Propriedade	Descrição
length	Número de URLs no histórico

Método	Descrição
back()	Volta para a URL anterior no histórico
forward()	Avança para a próxima URL no histórico
go(X)	Volta (valores negativos) ou avança (valores positivos) para uma URL da lista

¹ https://www.w3schools.com/jsref/obj_history.asp

Objeto navigator¹

Propriedade	Descrição
appName	Codigome do navegador
appName	Nome do navegador
appVersion	Versão do navegador
cookieEnabled	Verifica se cookies estão habilitados
geolocation ²	Objeto <i>geolocation</i> (coordenadas)
onLine	Verifica se o navegador está online

Método	Descrição
javaEnabled()	Verifica se Java está habilitado

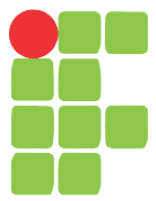
¹ https://www.w3schools.com/jsref/obj_navigator.asp

² https://www.w3schools.com/html/html5_geolocation.asp



Eventos

- ▶ São ações decorrentes de alguma interação com o usuário. Exemplos:
 - ▶ Carregar uma página (onload);
 - ▶ Clicar (onclick);
 - ▶ Passar o cursor do mouse sobre algum objeto (onmouseover);
 - ▶ Selecionar um campo em um formulário (onfocus).
- ▶ A partir de um evento, pode-se executar um script.



Principais eventos¹

Evento	Descrição
onload	Quando carregar a página
onfocus	Quando usuário “entra” no campo
onblur	Quando usuário “deixa” no campo
onchange	Quando usuário alterar o valor
onclick	Quando usuário clica

¹ https://www.w3schools.com/jsref/dom_obj_event.asp

Principais eventos (Mouse)

Evento	Descrição
onmouseover	Quando usuário “entra” o cursor do mouse sobre o elemento ou seus filhos
onmouseenter	Quando usuário “entra” o cursor do mouse sobre o elemento
onmouseout	Quando usuário move o cursor do mouse para fora do elemento ou de seus filhos
onmouseleave	Quando usuário move o cursor do mouse para fora do elemento
onmousemove	Quando usuário move o cursor do mouse sobre o elemento
onmousedown	Quando usuário pressiona o botão esquerdo do mouse
onmouseup	Quando usuário libera o botão esquerdo do mouse

Principais eventos (Teclado)

Evento	Descrição
onkeydown	Quando usuário pressiona uma tecla, disparado antes do “onkeypress”. Pode ser usado para evitar que a tecla pressionada seja escrita no campo
onkeypress	Quando usuário pressiona uma tecla, disparado após o “onkeydown”. Não é disparado para teclas que não geram caracteres (F1, F2, tab etc.), porém faz distinção entre letras maiúsculas e minúsculas
onkeyup	Quando usuário libera uma tecla

Objeto event¹

Propriedade	Descrição
type	Tipo do evento ocorrido
target	Objeto relacionado ao evento
which / keyCode ²	Código Unicode da tecla pressionada ou liberada
key	Tecla pressionada ou liberada
ctrlKey, altKey, shiftKey	Retorna se a tecla CTRL, ALT ou SHIFT foi pressionada
pageX, pageY	Coordenada X ou Y do cursor do mouse em relação à página

Método	Descrição
preventDefault()	Cancela o evento atual, caso seja cancelável

¹ https://www.w3schools.com/jsref/dom_obj_event.asp

² Alguns navegadores implementam o atributo which, outros, keyCode



Tipos de Dados, Variáveis e Operadores

Parte 03



Sumário

- ▶ Tipos de dados
- ▶ Variáveis
- ▶ Operadores



Tipos de dados

- ▶ JavaScript aceita os tipos: number (inteiros ou com decimais), string, boolean e null;
- ▶ Entretanto apresenta tipagem fraca, não sendo necessário declarar o tipo de uma variável e a mesma poderá receber valores de um tipo diferente, posteriormente.



Tipos de dados

Exemplo

```
var x = 0;
```

```
x = "Olá!";
```

```
x = true;
```

```
x = null;
```

Nesse exemplo, a variável `x` é declarada inicialmente com um valor numérico e, sucessivamente, atribui-se a ela uma string, um boolean e, por fim, `null`.



Tipos de dados

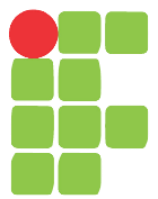
- ▶ Caso seja realizada operação entre string e outro tipo, a conversão dos dados dependerá do operador:
 - ▶ Operador suportado por string: converte não string em string;
 - ▶ Operador não suportado por string: converte string em não string.

Exemplos

"casa" + 1 // Resultado será "casa1"

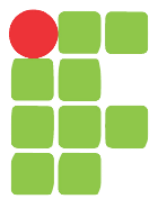
true + "bola" // Resultado será "truebola"

"5" * 4 // Resultado será 20



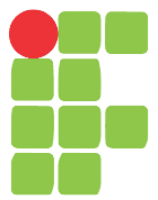
Funções para manipulação de strings e números

Função	Descrição	Exemplo
eval	Avalia a string passada	<pre>eval("2 + 2") eval("alert('Teste')")</pre>
parseInt	Converte string em inteiro	<pre>parseInt("12.3")</pre>
parseFloat	Converte string em número com decimais	<pre>parseFloat("12.3")</pre>
Number	Converte objeto em número	<pre>Number(true)</pre>
String	Converte objeto em string	<pre>String(true)</pre>
N.toString()	Converte o número em string	<pre>n.toString() (13.105).toString()</pre>
N.toFixed(X)	Converte o número em string com X casas decimais	<pre>n.toFixed(2) (13.105).toFixed(2)</pre>



Variáveis

- ▶ O identificador (nome) de uma variável segue a mesma regra de outras linguagens, isto é, deve começar por uma letra ou `_`, os caracteres seguintes podem ser letras, números ou `_` e não pode haver espaços ou caracteres especiais.



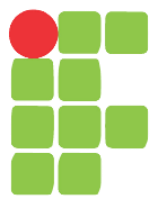
Declaração de variáveis

- ▶ Uma variável pode ser declarada usando a palavra reservada “var” ou fazendo-se uma atribuição a ela;
- ▶ Sem “var” (ex: `x = 1;`)
 - ▶ O escopo da variável é global, isto é, ela existirá ao longo de toda a página;
- ▶ Com “var” (ex: `var x;` ou `var x = 1;`)
 - ▶ Caso esteja dentro de uma função, seu escopo é local, isto é, só existirá dentro daquela função.



Operadores

- ▶ A maioria dos operadores de JavaScript são os mesmos de outras linguagens que seguem o padrão ECMA-262 (Java, ActionScript, PHP, C/C++ etc.);
- ▶ A seguir, listagem com os principais operadores por grupo e exemplos.



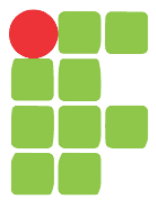
Operadores aritméticos

Operador	Significado	Exemplo
+	Soma	$5 + 2$
++	Incrementa uma unidade	<code>i++</code>
-	Subtração ou inversão de sinal	$5 - 2$ -3
--	Decrementa uma unidade	<code>i--</code>
*	Multiplicação	$5 * 2$
/	Divisão	$5 / 2$
%	Módulo (resto)	$5 \% 2$



Atribuições

Operador	Significado	Exemplo
=	Recebe (armazena um valor)	$x = 3$
+=	Recebe o valor dele adicionado com	$x += 3$
-=	Recebe o valor dele subtraído de	$x -= 3$
*=	Recebe o valor dele multiplicado por	$x *= 3$
/=	Recebe o valor dele dividido por	$x /= 3$
%=	Recebe o resto da divisão dele por	$x \% = 3$



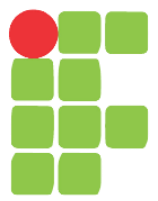
Operadores relacionais

Operador	Significado	Exemplo
>	Maior	5 > 2
<	Menor	5 < 2
>=	Maior ou igual	5 >= 2
<=	Menor ou igual	5 <= 2
==	Igual	5 == 5 5 == "5"
!=	Diferente	5 != 2
===	Igual e do mesmo tipo	5 === "5" 5 === 5



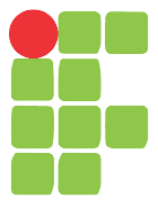
Operadores lógicos

Operador	Significado	Exemplo
&&	E	$x \ \&\& \ y$
	Ou	$X \ \ y$
!	Não	$!x$



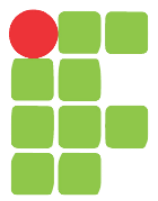
Operadores para string

Operador	Significado	Exemplo
+	Concatenação	$x + y$



Operadores especiais

Operador	Significado	Exemplo
?:	Segundo uma condição, retorna um valor ou outro	a ? b : c
in	Verifica se a propriedade especificada está no objeto especificado	title in plIntroducao
instanceof	Verifica se o objeto especificado é daquele tipo	x instanceof String
typeof	Retorna uma string indicando o tipo do operador	typeof "teste"
new	Permite criar um objeto a partir de um construtor	new String("teste")



Funções e comandos de controle e de repetição

Parte 04



Sumário

- ▶ Criando funções
- ▶ Comandos de controle
- ▶ Comandos de repetição



Criando funções

- ▶ Uma função é uma rotina (conjunto de instruções), geralmente composta por um cabeçalho (assinatura) e um corpo;

Sintaxe:

```
function nomeFuncao(argumentos) {  
    comandos;  
}
```



Exemplos de funções



```
//Muda cor de fundo da página  
function mudaCor(cor) {  
    document.backgroundColor = cor;  
}
```



Exemplos de funções



//Calcula e retorna a média de dois números

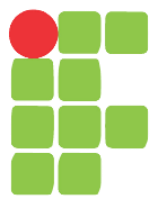
```
function calcMedia(a, b) {  
    return (a + b) / 2;  
}
```




Exemplos de funções

//Valida se campo foi preenchido

```
function validaCampo(campo) {  
  if (campo.value == "") {  
    alert("Você deve preencher o campo ao lado!");  
    return false;  
  }  
}
```



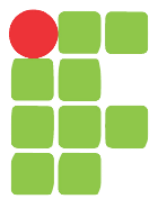
Comandos de controle de fluxo

if ... else

Se sua condição for verdadeira, executa um comando; Caso contrário (e se houver cláusula else), executa outro comando.

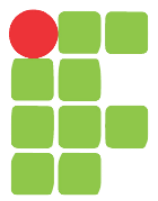
switch

Executa o bloco de comandos associado à constante que apresentar valor igual ao da expressão passada.



if ... else - Sintaxe

```
if (condicao) {  
    comandos;  
} else {  
    comandos;  
}
```



if ... else - Exemplo

```
//Exibe alerta dizendo se o número é par ou não  
if (numero % 2 == 0) {  
    alert(numero + “ é par!”);  
} else {  
    alert(numero + “ é ímpar!”);  
}
```



switch - Sintaxe

```
switch (expressão) {  
    case CONSTANTE:  
        comandos;  
        break;  
    default:  
        comandos;  
        break;  
}
```



switch - Exemplo

//Dado número do mês, exibe alerta com mês em texto

```
switch (mes) {  
    case 1: alert("Janeiro"); break;  
    case 2: alert("Fevereiro"); break;  
    case 3: alert("Março"); break;  
    ...  
}
```



Comandos de repetição

while

Enquanto sua condição for verdadeira, executa um comando.

for

Após inicializar um contador, até que a condição passada seja falsa, executa o comando associado.

for...in

Para cada propriedade de um objeto, executa o comando associado.



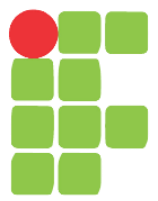
while - Sintaxe

```
while (condicao) {  
    comandos;  
}
```



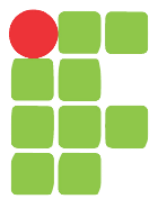

while - Exemplo

```
//Gera string contendo os x primeiros números pares  
var i = 0;  
var resposta = "";  
while (i < x) {  
    resposta = resposta + 2*i + ", ";  
    i++;  
}
```



for - Sintaxe

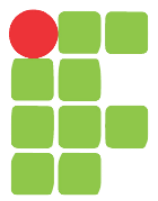
```
for (inicialização; condicao; incremento) {  
    comandos;  
}
```



for - Exemplo



```
//Gera string contendo os x primeiros números pares
for (var i = 0; i < x; i++) {
    resposta = resposta + 2*i + “, ”;
}
```



for...in - Sintaxe

```
for (var propriedade in objeto) {  
    comandos;  
}
```

propriedade - string com nome da propriedade;
objeto[propriedade] - valor da propriedade do objeto.



for...in - Exemplo



//Imprime as propriedades do objeto usuario

```
var usuario = {nome: "Christiano",  
              matricula: "001",  
              curso: "Informática"};  
for (var prop in usuario) {  
    document.writeln(prop, " -> ", usuario[prop], "<br>");  
}
```



Comandos de interrupção

break

Interrompe a iteração atual e finaliza o laço.

continue

Interrompe a iteração atual, mas não finaliza o laço (passando para a próxima iteração).



Objetos e Arrays

Parte 05



Sumário

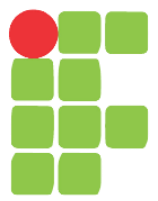
- ▶ Objetos
- ▶ Arrays



Objeto

- ▶ Trata-se de um variável composta heterogênea cujos valores podem ser armazenados ou recuperados a partir de um nome (propriedade).

Java	JavaScript
Quantidade de atributos imutável	Quantidade de atributos (propriedades) variável
Tipagem de cada atributo estática	Tipagem de cada atributo dinâmica



Declarando Objetos

Sintaxe #1

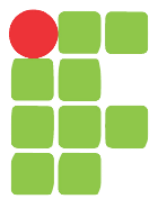
```
variavel = new Object( valor ); //Valor é opcional
```

Exemplo

```
aluno = new Object( "Joaquim" );
```

```
carro = new Object();
```

```
diario = new Object; //Não recomendado!
```



Declarando Objetos

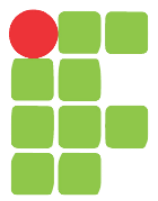
Sintaxe #2

```
variavel = { propriedade: valor, ... };
```

Exemplo

```
aluno = {mat: 1, nome: "Joaquim", idade: 19 };
```

```
carro = {};
```



Atribuindo valores às propriedades

Sintaxe #1

```
variavel.propriedade = valor; //Recomendado
```

Exemplo

```
aluno.nome = "Zacarias";
```



Atribuindo valores às propriedades

Sintaxe #2

```
variavel["propriedade"] = valor;
```

Exemplo

```
aluno["nome"] = "Zacarias";
```



Definindo métodos

Sintaxe #1

```
variavel.metodo = function () {...}; //Recomendado
```

Exemplo

```
aluno.gritar = function () { alert("Ahhh!") }
```



Definindo métodos

Sintaxe #2

```
variavel["método"] = function () {...};
```

Exemplo

```
aluno["gritar"] = function () { alert("Ahhh!") }
```



Array

- ▶ Trata-se de um variável composta cujos valores podem ser armazenados ou recuperados a partir de um índice.

Java	JavaScript
Tamanho imutável	Tamanho variável
Índice numérico	Índice numérico ou textual
Elementos homogêneos	Elementos não homogêneos (não há tipagem estática)



Declarando Arrays

Sintaxe #1

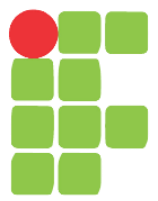
```
variavel = new Array(tamanho); //Tamanho é opcional
```

Exemplo

```
diasDaSemana = new Array(7);
```

```
meses = new Array();
```

```
anos = new Array; //Não recomendado!
```



Declarando Arrays

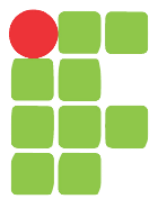
Sintaxe #2

```
variavel = [elemento1, elemento2... ];
```

Exemplo

```
diasDaSemana = ["Segunda", "Terça", "Quarta"];
```

```
meses = [];
```



Atribuindo valores ao Array

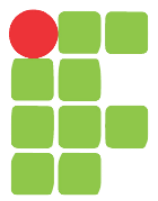
Sintaxe

```
variavel[indice] = valor;
```

Exemplo

```
meses[0] = "Janeiro";
```

```
notas["Joaquim"] = 10.0;
```



Propriedades de um Array

Propriedade	Descrição
length	Retorna o tamanho do array



Métodos de um Array

Método	Descrição
<code>concat(array2)</code>	Retorna um novo array concatenando os elementos deste com os de array2
<code>indexOf(elemento)</code>	Retorna a posição do elemento dentro do array
<code>push(elem1, elem2...)</code>	Adiciona novos elementos ao final do array, retornando o novo tamanho
<code>pop()</code>	Remove e retorna o último elemento do array
<code>unshift(elem1, elem2...)</code>	Adiciona novos elementos ao início do array, retornando ao novo tamanho
<code>shift()</code>	Remove e retorna o primeiro elemento do array
<code>toString()</code>	Retorna o array como String



Object, Date, Math e String

Parte 06



Sumário

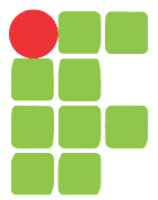
- ▶ Object
- ▶ Date
- ▶ Math
- ▶ String

SAIBA MAIS!

Object

Propriedade	Descrição
Object.prototype	Representa protótipo de Object
Object.length	Tamanho de Object (é sempre 1)

Método	Descrição
Object.keys(obj)	Retorna um array contendo os nomes de todas as propriedades de obj
Object.values(obj)	Retorna um array contendo os valores de todas as propriedades de obj
Object.entries(obj)	Retorna um array contendo as propriedades de obj no formato [nome, valor]



Date

- ▶ Permite armazenar objetos do tipo Date (data e hora) em variáveis;

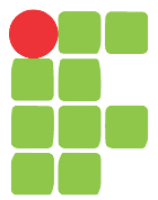
Exemplos

//Cria um objeto com a data atual

```
hoje = new Date();
```

//Ano, mês, dia, horas, minutos, segundos

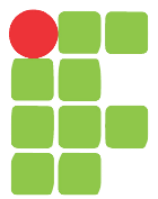
```
natal = new Date(2017, 11, 25, 5, 0, 0);
```



Date

Método	Descrição
getDate()	Retorna o dia do mês
getMonth()	Retorna o mês (0 para janeiro)
getFullYear()	Retorna o ano, com 04 dígitos
getHours()	Retorna as horas
getMinutes()	Retorna os minutos
getSeconds()	Retorna os segundos
getDay()	Retorna o dia da semana (0 para domingo)
toString()	Retorna a data como String

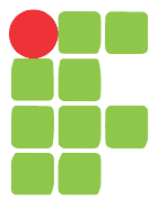
Método	Descrição
setDate(x)	Altera o dia do mês
setMonth(x)	Altera o mês
setFullYear(x)	Altera o ano
setHours(x)	Altera as horas
setMinutes(x)	Altera os minutos
setSeconds(x)	Altera os segundos



Math

Propriedade	Descrição
Math.E	Retorna a constante de Euler (aprox. 2,718)
Math.PI	Retorna a constante PI (aprox. 3,14)

Método	Descrição
Math.abs(x)	Retorna o valor absoluto de um número
Math.ceil(x)	Retorna o número inteiro arredondado para cima
Math.floor(x)	Retorna o número inteiro arredondado para baixo
Math.max(num1, ...)	Retorna o maior número de uma série
Math.min(num1, ...)	Retorna o menor número de uma série
Math.random()	Retorna um número aleatório decimal entre 0 e 1
Math.round(x)	Retorna o valor do número arredondado



String

Propriedade	Descrição
length	Retorna o tamanho da String

Método	Descrição
charAt(indice)	Retorna o caracter no índice especificado
indexOf(substring)	Retorna a posição de substring dentro da String
replace(antigo, novo)	Substitui a ocorrência da substring antigo pela substring novo
slice(inicio, fim)	Retorna uma substring, começando na posição inicio e terminando na posição fim
split(separador, limite)	“Quebra” a String baseada no caracter separador, retornando-a como um array de substrings com um máximo de elementos especificado por limite
toLowerCase()	Retorna a String em minúsculas
toUpperCase()	Retorna a String em maiúsculas



Introdução ao Bootstrap

Parte 07



Sumário

- ▶ O que é Bootstrap?
- ▶ Exemplos de regras CSS presentes no Bootstrap
- ▶ Como baixar o Bootstrap?
- ▶ Estrutura de arquivos
- ▶ Como usar em uma página HTML?
- ▶ Estrutura básica de uma página
- ▶ Estrutura de página com três seções



O que é Bootstrap?

- ▶ É um *framework* para desenvolvimento *front-end* de websites, liberado inicialmente pelo Twitter;
- ▶ Usa HTML, CSS e JavaScript para criar uma experiência de usuário segundo a filosofia *mobile-first*;
- ▶ Define um conjunto de classes CSS e funções em JS para agilizar o desenvolvimento de websites responsivos;
- ▶ Versão atual: v4.1.3.



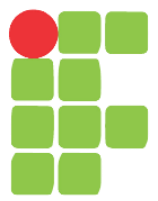
Exemplos de regras CSS presentes no Bootstrap

```
hr {  
    box-sizing: content-box;  
    height: 0;  
    overflow: visible;  
}
```




Exemplos de regras CSS presentes no Bootstrap

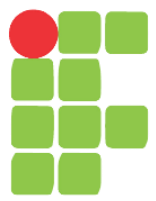
```
h1, h2, h3, h4, h5, h6 {  
  margin-top: 0;  
  margin-bottom: 0.5rem;  
}
```



Como usar o Bootstrap?

- ▶ Opção CDN
 - ▶ Aponte os links para servidores CDN com cada arquivo;
- ▶ Opção *Hosted*
 - ▶ Acesse <http://getbootstrap.com>
 - ▶ Clique em “Download”
 - ▶ Baixe a versão compilada (“Compiled CSS and JS”)
 - ▶ Inclua os arquivos em seu website

Vamos optar pelo uso de arquivos em CDN.



Como usar em uma página HTML?

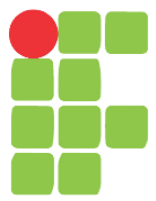
► Inclua na seção HEAD de uma página HTML 5 as seguintes tags:

```
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
">
```

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
```

```
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"><
/script>
```



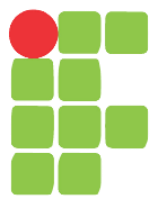
Como usar em uma página HTML?

- ▶ Inclua na seção BODY de uma página HTML 5 uma tag com a classe “container” ou “container-fluid”:

```
<div class="container">
```

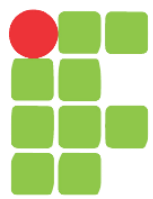
Seu conteúdo vai aqui!

```
</div>
```



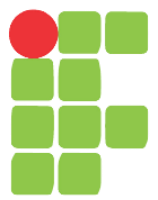
Como usar em uma página HTML?

- ▶ **container** - especifica que a largura do container será fixa, mudando apenas quando a largura da tela atinge um *breakpoint*;
- ▶ **container-fluid** - especifica que o container ocupará sempre 100% da largura disponível.



Estrutura básica de uma página

```
<!DOCTYPE html>
<html>
  <head>
    <title>Título vai aqui!</title>
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
  </head>
  <body>
    <div class="container" style="background-color:#AFF">
      <p><i><b>Desafio:</b> Qual a diferença entre usar a classe container e a classe container-fluid?</p>
    </div>
  </body>
</html>
```



Estrutura de página com três seções

- ▶ Seções:
 - ▶ Nav
 - ▶ Main
 - ▶ Footer

Cursos EAD

Seja bem-vindo!

Uma mensagem de boas vindas deveria aparecer aqui!

© ChristianoSantos.com 2017



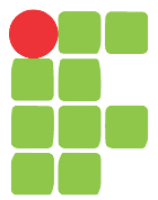
Ícones, Breakpoints e Sistema de Grid

Parte 08



Sumário

- ▶ Ícones
- ▶ Breakpoints e visibilidade dos elementos
- ▶ Sistema de Grid
- ▶ Exemplos de página com três colunas



Ícones

8.1

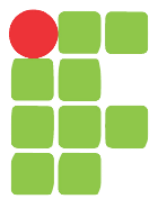


- ▶ Usaremos o pacote gratuito da Font Awesome (<http://fontawesome.io>), que pode ser usado para fins comerciais ou não;
- ▶ Para usar um ícone:
 - ▶ Certifique-se de associar seu CSS corretamente:

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
```
 - ▶ Inclua o ícone usando a tag `` e sua classe correspondente:

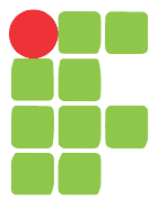
```
<span class="fa fa-book"></span>
```
 - ▶ Pode-se alterar seu tamanho usando a propriedade `font-size`:

```
<span class="fa fa-book" style="font-size:3.0rem"></span>
```



Breakpoints e visibilidade dos elementos

- ▶ A fim de definir como uma tag deve ser exibida para cada tamanho de tela, Bootstrap emprega *breakpoints*;
- ▶ Cada *breakpoint* representa um valor a partir do qual a tela é grande o suficiente para que o layout seja tratado de forma diferente;
- ▶ São parte dos nomes das tags.



Breakpoints e visibilidade dos elementos

Breakpoint	Tamanho de tela alvo
xs	Aplica-se a dispositivos com telas muito pequenas, tais como smartphones com largura menor que 576px
sm Ex: <code>.d-sm-none</code>	Aplica-se a dispositivos com telas pequenas, tais como phablets e tablets com largura entre 576px e 768px
md Ex: <code>.d-md-block</code>	Aplica-se a dispositivos com telas médias, tais como desktops e notebooks com largura entre 768px e 992px
lg Ex: <code>.d-lg-inline</code>	Aplica-se a dispositivos com telas grandes, tais como grandes monitores desktop e TV com largura entre 992px e 1200px
xl Ex: <code>.d-xl-inline</code>	Aplica-se a dispositivos com telas muito grandes, tais como TVs com largura maior que 1200px



Breakpoints e visibilidade dos elementos

Exemplos de uso:

```
<div class="d-block"></div>
```

```
<div class="d-block d-md-none"></div>
```

```
<div class="d-none d-lg-block"></div>
```



Breakpoints e visibilidade dos elementos

Exemplos de uso:

```
<div class="d-none d-sm-block"></div>
```

```
<div class="d-block d-lg-inline"></div>
```

Renderizando imagens

- ▶ A fim de facilitar layouts responsivos, recomenda-se:
 - ▶ Não especificar o tamanho exato das imagens em pixels;
 - ▶ Utilizar a classe `img-fluid` ou `img-thumbnail`;
 - ▶ Pode usar também a classe `rounded` ou `rounded-circle`.

```
.img-fluid {  
    max-width: 100%;  
    height: auto;  
}
```

```
.img-thumbnail {  
    padding: 0.25rem;  
    border: 1px solid #ddd;  
    border-radius: 0.25rem;  
    max-width: 100%;  
    height: auto;  
}
```



Sistema de Grid

- ▶ A fim de facilitar a criação de layouts, o Bootstrap utiliza-se de um sistema de grid que “divide” a tela em 12 colunas;
- ▶ Assim, ao inserir um novo elemento no HTML, pode-se especificar o espaço que ocupará no layout pela “quantidade de colunas” que ocupará;
- ▶ Sintaxe: `.col-{breakpoint}-{1..12}`



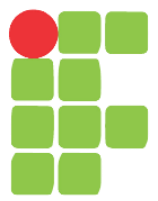
Sistema de Grid

Exemplos:

```
<div class="col-10">
```

Esta div ocupará 10 colunas, sempre

```
</div>
```



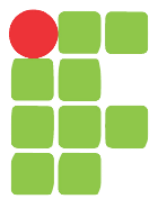
Sistema de Grid

Exemplos:

```
<div class="col-sm-2">
```

Se o dispositivo é "sm" ou maior, esta div ocupará
2 colunas

```
</div>
```



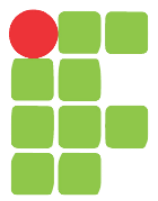
Sistema de Grid

Exemplos:

```
<div class="col-md-4">
```

Se o dispositivo é "md" ou maior, esta div ocupará 4 colunas

```
</div>
```



Sistema de Grid

Exemplos:

```
<div class="col-lg-6">
```

Se o dispositivo é "lg" ou maior, esta div ocupará 6 colunas

```
</div>
```



Sistema de Grid

Exemplos:

```
<div class="col-10 col-sm-2">
```

Esta div ocupará 10 columnas, mas se o dispositivo for “sm”, ocupará 2

```
</div>
```



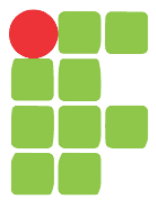
Exemplo de página com três colunas

```
<div class="row">  
  <div class="col-2"></div>  
  <div class="col-8"></div>  
  <div class="col-2"></div>  
</div>
```



Exemplo de página com três colunas (layout mais responsivo)

```
<div class="row">  
  <div class="col-sm-2"></div>  
  <div class="col-sm-10 col-md-8"></div>  
  <div class="col-md-2"></div>  
</div>
```



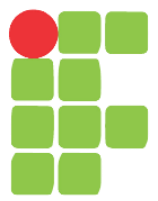
Nosso exemplo de site de cursos online!

Agora, nossa página possui uma *sidebar* que, em telas pequenas, ajusta-se abaixo do conteúdo principal!



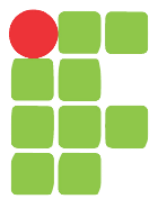
Alguns Componentes no Bootstrap

Parte 09



Sumário

- ▶ Barra de navegação
- ▶ Cards
- ▶ Jumbotron
- ▶ Badges
- ▶ Alertas
- ▶ Barra de ferramentas
- ▶ Breadcrumb Navigation
- ▶ Paginação



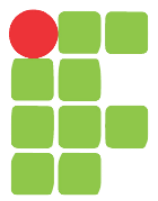
Barra de navegação

- ▶ Muitos *web apps* e *mobile apps* atuais apresentam uma barra de navegação na parte superior (ou inferior) da tela, contendo o nome da mesma (“marca”) e as opções/seções mais importantes;
- ▶ Em HTML 5, geralmente usamos a tag `<nav>` para criá-las.

```
<nav>
```

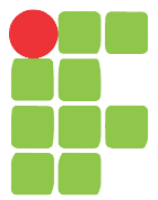
```
    Conteúdo aqui!
```

```
</nav>
```



Barra de navegação

- ▶ Em Bootstrap, algumas classes úteis para criação da barra de navegação:
 - ▶ **navbar** - identifica o elemento como sendo uma barra de navegação;
 - ▶ **navbar-expand-{breakpoint}** - determina que todos os itens da barra fiquem um ao lado do outro (alinhados à esquerda);
 - ▶ **fixed-top** / **fixed-bottom** - informa que o elemento deve estar fixo ao topo / fundo;
 - ▶ **bg-dark** / **bg-light** / **bg-white** / **bg-transparent** - usa a cor de background escura / clara / branca / transparent.



Barra de navegação

Exemplo:

```
<nav class="navbar navbar-expand-md fixed-top navbar-dark bg-dark">
```

Opções vão aqui!

```
</nav>
```

Obs: Caso utilize a classe **fixed-top**, lembre-se que a barra de navegação ficará sobre a página, fora do fluxo normal, então precisará adicionar um espaçamento no topo do body (**padding-top**).

De forma análoga, se usar a classe **fixed-bottom**, adicione espaçamento no final do body (**padding-bottom**).



Barra de navegação

- ▶ Pode-se inserir um hiperlink estilizado para representar a marca do website, usando-se a classe **navbar-brand**.

```
<a class="navbar-brand" href="index.html">  
  Cursos EAD  
</a>
```



Barra de navegação

- ▶ Para inserir opções na barra de navegação, podemos usar uma lista não-ordenada com as devidas classes CSS:
 - ▶ **navbar-nav**, para a tag ``;
 - ▶ **nav-item**, para as tags ``;
 - ▶ **active**, para a tag `` ativa no momento;
 - ▶ **nav-link**, para as tags `<a>`.



Barra de navegação

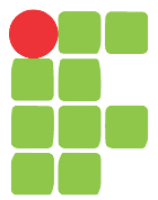
```
<ul class="navbar-nav">
  <li class="nav-item active">
    <a class="nav-link" href="index.html">Home</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="cursos.html">Todos os cursos</a>
  </li>
</ul>
```




Barra de navegação

- ▶ Para ocultar as opções da *navbar* quando a tela é pequena, precisamos inseri-las em um div como o seguinte:

```
<div id="navbarCollapse" class="navbar-collapse collapse">  
  Lista de opções aqui!  
</div>
```



Barra de navegação

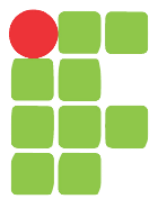
- ▶ E agora, precisamos de um botão (antes da div) para exibi-las novamente:

```
<button class="navbar-toggler" data-toggle="collapse"
        data-target="#navbarCollapse">
  <span class="navbar-toggler-icon"></span>
</button>
```

Obs: Deve-se incluir o script

<https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js>

a fim de o botão funcionar corretamente.



Barra de navegação com Dropdown

- ▶ Caso deseje um menu dropdown em sua barra de navegação, você utilizará as seguintes classes:
 - ▶ dropdown - na tag li que se expandirá;
 - ▶ dropdown-toggle - na tag a que receberá o clique para expandir;
 - ▶ dropdown-menu - na tag div contendo as opções a serem exibidas;
 - ▶ dropdown-item - em cada tag a de cada opção.
- ▶ Além disso, ao link que será clicado para exibir as opções, devemos acrescentar o atributo *data-toggle="dropdown"*, necessário para o funcionamento correto.

Barra de navegação com Dropdown

- ▶ Alterando o exemplo anterior para conter três opções de cursos em “Todos os cursos” :

```
<li class="nav-item dropdown">  
  <a class="nav-link dropdown-toggle" href="#" data-toggle="dropdown">Todos  
os cursos</a>  
  <div class="dropdown-menu">  
    <a class="dropdown-item" href="#">Programação</a>  
    <a class="dropdown-item" href="#">Arte Gráfica</a>  
    <a class="dropdown-item" href="#">Música</a>  
  </div>  
</li>
```



Cards

- ▶ Um card (“cartão”) permite a criação de um painel, um container delimitado por bordas que pode conter uma área para cabeçalho, conteúdo e rodapé.



Cards

Exemplo de card simples

Conteúdo aqui!

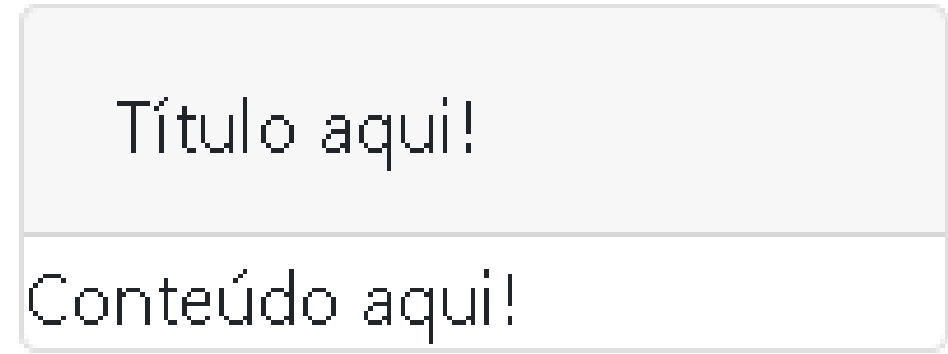
```
<div class="card">  
  Conteúdo aqui!  
</div>
```



Cards

Exemplo de card com cabeçalho

```
<div class="card">  
  <div class="card-header">  
    Título aqui!  
  </div>  
  Conteúdo aqui!  
</div>
```





Cards

9.9



Exemplo de card com cabeçalho e rodapé

```
<div class="card">  
  <div class="card-header">Título aqui!</div>  
  Conteúdo aqui!  
  <div class="card-footer">Rodapé aqui!</div>  
</div>
```

Título aqui!

Conteúdo aqui!

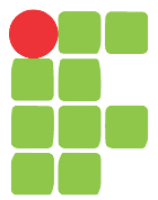
Rodapé aqui!



Cards



Vejam como fica em nossa página...



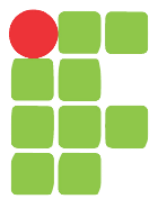
Jumbotron

- ▶ Trata-se de um componente que pode capturar toda a viewport para exibir algum conteúdo.

```
<div class="jumbotron">  
  Conteúdo aqui!  
</div>
```

Página não encontrada!

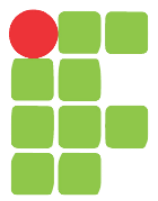
[Ir para a página inicial](#)



Jumbotron



- ▶ O jumbotron pode ser usado fora de um container...



Jumbotron



► ...ou dentro de um.



Jumbotron



- ▶ Vamos usar um para exibir um anúncio promocional em nossa página.



Badges

- ▶ Um *badge* é um indicador formatado usado para exibir uma contagem de itens (itens em um carrinho, número de e-mails novos, notificações não abertas etc.);
- ▶ Para exibir um *badge* na página:
`Valor`



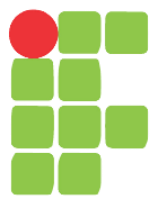
Badges



Exemplo de badge para carrinho de compras

```
<a href="carrinho.html">  
  <span class="fa fa-shopping-cart"></span>  
  <span class="badge badge-primary">4</span>  
</a>
```





Badges

Adicionando badge à barra de navegação

```
<li class="nav-item">  
  <a class="nav-link" href="#">  
    <span class="fa fa-envelope"></span>  
    Notificações  
    <span class="badge badge-secondary">4</span>  
  </a>  
</li>
```

 Cursos EAD

Home

Todos os cursos

Contato

 Notificações 4



Badges



Podemos adicioná-lo à outra tag ul (na mesma navbar) para colocar nossa notificação à direita...

```
<ul class="navbar-nav mr-auto">
```

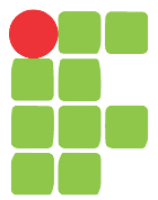
```
...
```

```
</ul>
```

```
<ul class="navbar-nav">
```

```
...
```

```
</ul>
```



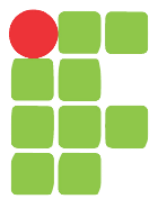
Alertas

- ▶ Caso queira exibir uma seção destacada na página com uma mensagem ao usuário, pode-se utilizar a classe `alerta` com uma das opções de contexto (`alert-info`, `alert-warning`, `alert-success`, `alert-danger`);

- ▶ Exemplo:

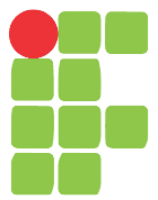
```
<div class="alert alert-info">Mensagem</div>
```

Mensagem



Barra de ferramentas

- ▶ Para agrupar alguns botões em uma barra de ferramentas:
 - ▶ Usar classe "btn-group" na tag pai/container;
 - ▶ Usar classe "btn" em cada um deles, bem como a classe que melhor define seu contexto (ex: btn-success, btn-warning, btn-danger, btn-default etc.).



Barra de ferramentas

Exemplo:

```
<div class="btn-group">  
  <button type="button" class="btn btn-success">Novo</button>  
  <button type="button" class="btn btn-warning">Editar</button>  
  <button type="button" class="btn btn-danger">Excluir</button>  
</div>
```





Breadcrumb Navigation

- ▶ A breadcrumb navigation (navegação em “fatias”) permite listar de forma intuitiva o caminho que vai da página inicial até a página atual passando por cada nível;
- ▶ Para tal, basta criarmos uma lista ordenada com a classe “breadcrumb” e listar cada hyperlink como um item de lista.



Breadcrumb Navigation

Exemplo:

```
<ol class="breadcrumb">  
  <li class="breadcrumb-item"><a href="#">Home</a></li>  
  <li class="breadcrumb-item"><a href="#">Informática</a></li>  
  <li class="breadcrumb-item"><a href="#">Programação</a></li>  
  <li class="breadcrumb-item active">Java</li>  
</ol>
```

[Home](#) / [Informática](#) / [Programação](#) / [Java](#) / Programação em Java, Deitel & Deitel



Paginação

- ▶ Para criar um componente de paginação:
 - ▶ Crie uma lista não-ordenada com a classe “pagination”;
 - ▶ Para cada item da lista, aplique a classe “page-item”;
 - ▶ Cada item conterá um hyperlink com a classe “page-link”.



Paginação

Exemplo:

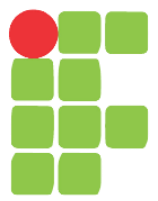
```
<ul class="pagination">  
  <li class="page-item"><a class="page-link" href="#">&laquo;</a></li>  
  <li class="page-item"><a class="page-link" href="#">1</a></li>  
  <li class="page-item"><a class="page-link" href="#">2</a></li>  
  <li class="page-item"><a class="page-link" href="#">3</a></li>  
  <li class="page-item"><a class="page-link" href="#">&raquo;</a></li>  
</ul>
```





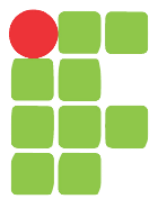
Formatando tabelas e formulários

Parte 10



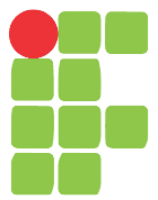
Sumário

- ▶ Formatando tabelas
- ▶ Formatando formulários



Formatando tabelas

- ▶ Treinemos agora algumas opções para a formatação de tabelas em Bootstrap. Inicie criando uma tabela apenas com as tags `table`, `thead`, `tbody`, `tr`, `td` e `th` como a seguinte.



Formatando tabelas

```
<table>
  <thead>
    <tr><th>Aluno</th><th>Nota</th></tr>
  </thead>
  <tbody>
    <tr><td>Antônio</td><td>10,0</td></tr>
    <tr><td>Joaquim</td><td>10,0</td></tr>
    <tr><td>Maria</td><td>10,0</td></tr>
    <tr><td>Zacarias</td><td>10,0</td></tr>
  </tbody>
</table>
```



Formatando tabelas

10.2

- ▶ Agora, aplique cada uma das seguintes classes à tag table (de forma acumulada) e descreva o resultado:
 - ▶ table
 - ▶ table-bordered
 - ▶ table-striped
 - ▶ table-hover
 - ▶ table-responsive



Formatando tabelas

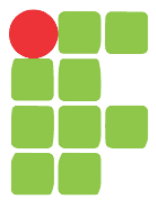
10.3

- ▶ Agora, experimente as opções de cores (classes) `thead-light` e `thead-dark` para a tag `thead`.



Formatando formulários

- ▶ Agora vamos treinar como formatar um formulário para ficar mais agradável visualmente;
- ▶ Como criávamos um formulário básico de contato em HTML 5 com os campos nome, e-mail, mensagem e botão para submissão?



Formatando formulários (parte 1)

```
<form>
```

```
  <div>
```

```
    <label for="seuNome">Nome</label>
```

```
    <input type="text" id="seuNome" placeholder="Seu nome, por favor">
```

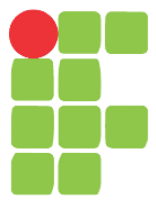
```
  </div>
```

```
  <div>
```

```
    <label for="seuEmail">E-mail</label>
```

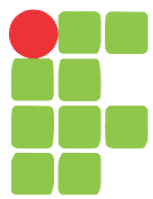
```
    <input type="email" id="seuEmail" placeholder="Seu e-mail">
```

```
  </div>
```

Formatando formulários (parte 2)

```
<div>  
  <label for="suaMensagem">Mensagem</label>  
  <textarea id="suaMensagem" placeholder="Mensagem"  
rows="3"></textarea>  
</div>  
<button type="submit">Enviar</button>  
</form>
```



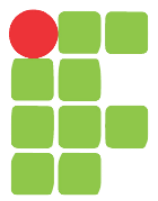
Formatando formulários

- ▶ Agora, aplique as seguintes classes:
 - ▶ Às tags div, classe “form-group row”;
 - ▶ Às tags label, classe “col-sm-2”;
 - ▶ A todas as tags digitáveis, classe “form-control”;
 - ▶ À tag button, classes “btn btn-primary”.

- ▶ Envolver cada input/textarea com um div contendo a classe “col-sm-10 input-group”.

Formatando formulários

- ▶ Inclua antes de cada input/textarea (dentro do div) um dos seguintes spans:
 - ▶ ``
 - ▶ `@`
 - ▶ ``
- ▶ Envolver o botão com uma div com as classes “col-sm-12 text-center” e inclua o seguinte span antes do texto “Enviar”:
 - ▶ ``



Formatando formulários

Obs: Para ver um exemplo de página com formulário de login, veja o arquivo **bs-signin.html**, da pasta “exemplos”

(extraído de <http://getbootstrap.com>).



Alguns componentes mais avançados

Parte 11



Sumário

- ▶ Janela modal
- ▶ Carrossel
- ▶ Abas



Introdução

- ▶ Alguns componentes requerem conhecimentos em JavaScript e jQuery para que possa tirar melhor proveito do Bootstrap;
- ▶ Porém, mesmo sem o devido conhecimento ainda é possível tirar algum proveito, já que o Bootstrap nos oferece funções pré-definidas para o funcionamento de alguns componentes mais avançados.



Janela modal

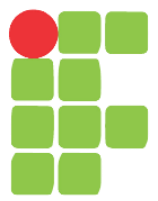
- ▶ Uma janela modal é um componente visual que se abre a página bloqueando toda a sua área;
- ▶ Geralmente aberta a partir de alguma ação do usuário e possui um botão para fechá-la;
- ▶ O Bootstrap oferece algumas classes para a criação desse tipo de janela.



Janela modal

- ▶ 1º passo: Crie dois div aninhados para representar a janela em si, como no exemplo abaixo:

```
<div class="modal fade" id="janelaModal">  
  <div class="modal-dialog modal-content">  
    ...  
  </div>  
</div>
```

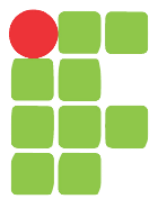


Janela modal

- ▶ 2º passo: Dentro da div mais interna, crie o cabeçalho, o corpo e o rodapé de sua janela.

Exemplo de cabeçalho:

```
<div class="modal-header">  
  <h4 class="modal-title">Título da janela</h4>  
  <button type="button" class="close" data-  
dismiss="modal">&times;</button>  
</div>
```



Janela modal

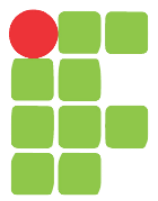
- ▶ 2º passo: Dentro da div mais interna, crie o cabeçalho, o corpo e o rodapé de sua janela. (cont.)

Exemplo de corpo:

```
<div class="modal-body">
```

```
    Conteúdo da janela.
```

```
</div>
```



Janela modal

- ▶ 2º passo: Dentro da div mais interna, crie o cabeçalho, o corpo e o rodapé de sua janela. (cont.)

Exemplo de rodapé:

```
<div class="modal-footer">  
  <button type="button" class="btn btn-secondary" data-dismiss="modal">  
    Close  
  </button>  
</div>
```



Janela modal

- ▶ 3º passo: Crie em sua página um botão que, ao ser clicado, abra a janela criada. Exemplo:

```
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#janelaModal">
```

Abrir Janela

```
</button>
```



Carrossel

- ▶ Um carrossel funciona como um container para a execução cíclica de imagens ou outras informações visuais (um “slideshow”);
- ▶ Um carrossel pode conter três conjuntos de elementos:
 - ▶ Indicadores, figuras que representam cada um dos “slides” no carrossel;
 - ▶ O slideshow em si, contendo os div a serem passados;
 - ▶ Os controles anterior e próximo.



Carrossel

- ▶ 1º passo: Crie o div que conterá o carrossel.
Exemplo:

```
<div id="demo" class="carousel slide" data-ride="carousel">
```

```
...
```

```
</div>
```



Carrossel

- ▶ 2º passo: Crie dentro do div uma lista não-ordenada contendo os indicadores. Exemplo:

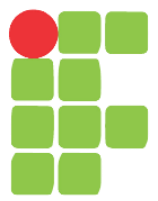
```
<ul class="carousel-indicators">  
  <li data-target="#demo" data-slide-to="0" class="active"></li>  
  <li data-target="#demo" data-slide-to="1"></li>  
</ul>
```




Carrossel

- ▶ 3º passo: Agora, crie um div contendo os elementos a serem passados (slideshow):

```
<div class="carousel-inner">  
  <div class="carousel-item active">  
      
  </div>  
  <div class="carousel-item">  
      
  </div>  
</div>
```

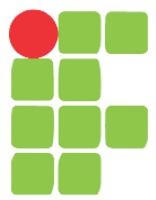


Carrossel



- ▶ 4º passo: Usando hyperlinks, crie os controles para retroceder ou avançar nos slides. Exemplo:

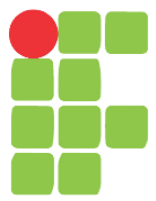
```
<a class="carousel-control-prev" href="#demo" data-slide="prev">  
  <span class="carousel-control-prev-icon"></span>  
</a>  
<a class="carousel-control-next" href="#demo" data-slide="next">  
  <span class="carousel-control-next-icon"></span>  
</a>
```



Carrossel

Obs: Caso deseje acrescentar algum rótulo/legenda ou outra informação a um dos “slides”, basta inserir um div com a classe “carousel-caption” contendo as informações desejadas. Veja como ficaria um dos itens do nosso exemplo:

```
<div class="carousel-item active">  
    
  <div class="carousel-caption">  
    <h3>Los Angeles</h3>  
    <p>We had such a great time in LA!</p>  
  </div>  
</div>
```



Precisa de mais componentes?

- ▶ Na página-tutorial sobre Bootstrap 4 da W3Schools, há vários outros componentes com exemplos de código:
<https://www.w3schools.com/bootstrap4/>



Criando uma galeria de imagens

Parte 12



Sumário

▶ Galeria de imagens



Como criar uma galeria de imagens?

- ▶ 1º passo: Incluir o CSS do BaguetteBox;

```
<link rel="stylesheet"  
href="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.8.1  
/baguetteBox.min.css">
```



Como criar uma galeria de imagens?

- ▶ 2º passo: Incluir a classe “gallery” na tag container das imagens da galeria;

```
<div class="gallery">
```

```
...
```

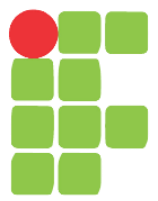
```
</div>
```




Como criar uma galeria de imagens?

- ▶ 3º passo: Incluir tags hyperlinks apontando para as imagens grandes com tags `img` exibindo as imagens pequenas (*thumbnails*);

```
<div class="gallery">  
  <a href="imagens/img01.jpg">  
      
  </a>  
  <a href="imagens/img02.jpg">  
      
  </a>  
</div>
```

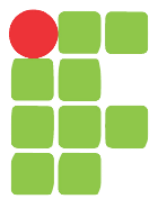


Como criar uma galeria de imagens?

- ▶ 4º passo: Em qualquer lugar da página após a galeria de imagens, carregue e execute o script do BaguetteBox;

```
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/baguettebox.js/1.8.1/baguetteBo  
x.min.js"></script>
```

```
<script>  
    baguetteBox.run('.gallery');  
</script>
```



Exemplo de galeria

► Disponível em:

<https://emtobias.com/semana-academica/2018/index.php>

```
<div class="gallery">  
  <div class="divGaleria">  
    <a href="imagens/img01.jpg">  
        
    </a>  
  </div>  
  ...  
</div>
```