



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SERGIPE

Campus Tobias Barreto

Construção de Sites

Prof. Me. Christiano Lima Santos

2020

Versão 1.0

Sumário

Parte 1 – HTML 5	9
Introdução à HTML 5	10
O que é HTML?.....	10
Evolução da web.....	10
Editores HTML.....	11
Estrutura Básica de um Documento HTML	12
Estrutura básica de uma tag.....	12
Estrutura básica de um documento HTML.....	12
Atributos globais mais comuns.....	13
Tag Head.....	13
Tag Body.....	13
Tags HTML Básicas	15
Tags Cabeçalhos (h1...h6).....	15
Tag Parágrafo (p).....	15
Tag Quebra de Linha (br).....	16
Tag Linha Horizontal (hr).....	16
Tag Hyperlink (a).....	17
Tags para Multimídia	19
Imagem.....	19
Áudio.....	20
Vídeo.....	20
Tags para Formatação	22
Tags para Negrito.....	22
Tags para Itálico.....	22
Tags para Sublinhado.....	22
Tag para Texto Pré-Formatado.....	23
Tags para subscrito e sobrescrito.....	23
Tag para marcado, deletado e inserido.....	23
Tabelas, Listas e Iframes	25
Tags para Tabelas.....	25
Tags para Listas.....	27
Tags para Iframes.....	28
Tags para Formulários	30
Tag Form.....	30
Tags Input, TextArea e Select	35
Tag Input.....	35
Alguns tipos de input.....	35
Outros atributos do input.....	36
Outros tipos de input.....	37
Tag TextArea.....	37
Tag Select.....	37
Tag Input com Datalist.....	38
Outras Tags Úteis	39

Tag Div.....	39
Tag Div – exemplo (a).....	39
Tag Div – exemplo (b).....	40
Tag Div – exemplo (c).....	40
Tag Div – exemplo (d).....	40
Tag Span.....	41
Tag Label.....	41
Tags Semânticas.....	41
Novas Tags Semânticas.....	42
Parte 2 – CSS 3.....	43
Introdução a CSS 3.....	44
O que é CSS?.....	44
Benefícios da adoção de CSS.....	44
Editores CSS.....	44
Como usar CSS em uma página.....	44
Sintaxe CSS.....	45
Seletores em CSS 3.....	46
Seletores.....	46
Tipos de Seletores.....	46
Pseudoclasses.....	47
Pseudoclasses mais usadas.....	48
Pseudoelementos.....	48
Combinação de Seletores.....	49
Exemplo: Uma página HTML, múltiplos estilos.....	49
Propriedades para Background e Borda.....	50
Propriedades para background.....	50
Propriedade background-color.....	50
Propriedade background-image.....	50
Propriedade background-size.....	51
Propriedade background-repeat.....	51
Propriedade background-position.....	51
Propriedade background-attachment.....	52
Propriedade background.....	52
Propriedades para borda.....	52
Propriedade border-style.....	53
Propriedade border-width.....	53
Propriedade border-color.....	53
Propriedade border-radius.....	53
Propriedade border.....	53
Propriedades para contorno.....	54
Propriedades outline-color, outline-offset, outline-style, outline-width.....	54
Propriedade outline.....	54
Propriedades para margem, padding, largura e altura.....	55
Propriedades para margem e padding.....	55
Propriedades para margem.....	55
Propriedades margin-top, margin-right, margin-bottom, margin-left.....	55
Propriedade margin.....	55
Propriedades para <i>padding</i>	56

Propriedades padding-top, padding-right, padding-bottom, padding-left.....	56
Propriedade padding.....	56
Propriedades para largura e altura.....	57
Propriedades width, min-width, max-width.....	57
Propriedades height, min-height, max-height.....	57
O “Modelo Caixa” em CSS.....	57
Largura e altura total de um elemento.....	58
Propriedades para textos, fontes e listas.....	59
Propriedades para textos.....	59
Propriedades color e line-height.....	59
Propriedades text-align e vertical-align.....	59
Propriedades text-indent e text-decoration.....	59
Propriedades para fontes.....	60
Propriedade font-family.....	60
Propriedade font-size.....	60
Propriedade font-style.....	61
Propriedade font-variant.....	61
Propriedade font-weight.....	61
Propriedade font.....	61
Propriedades para listas.....	61
Propriedade list-style-type.....	62
Propriedade list-style-image.....	62
Propriedade list-style-position.....	62
Propriedade list-style.....	62
Propriedades para layout e transparência.....	64
Propriedades para layout.....	64
Propriedade display.....	64
Propriedade float.....	64
Propriedade clear.....	64
Propriedades para transparência.....	64
Propriedade opacity.....	65
Propriedade background-color aplicando rgba.....	65
Outras propriedades para layout.....	66
Propriedade visibility.....	66
Propriedade position.....	66
Propriedade z-index.....	66
Barras de navegação em CSS.....	68
Criando uma barra de navegação (passo-a-passo).....	68
Barra de navegação com menu dropdown (passo-a-passo).....	69
Exemplos.....	71
Parte 3 – Boas Práticas para a Web.....	73
Otimização para motores de busca.....	74
Otimização para motores de busca (SEO).....	74
Fatores on-page e off-page a considerar.....	74
Guia para iniciantes (da Google).....	74
Outras dicas.....	75
O que evitar.....	76

Princípios de Web Design.....	77
Princípios de Design.....	77
Alinhamento.....	77
Balanço.....	77
Proximidade.....	78
Contraste.....	78
Repetição / Consistência.....	79
Princípios de Web Design.....	80
Usabilidade.....	80
Ergonomia.....	80
Acessibilidade.....	80
Tipos de Websites.....	81
Minisites.....	81
Hotsites.....	81
Sites para entretenimento.....	81
Sites educacionais.....	81
Blogs, vlogs e flogs.....	81
Catálogos, lojas virtuais, sites afiliados e marketplace.....	81
Sites afiliados.....	81
Sites portfólios.....	82
Sites institucionais.....	82
Sites comunitários.....	82
Sites midiáticos.....	82
Sites aplicativos.....	82
Sites bancos de dados.....	82
Portais.....	82
Parte 4 - JavaScript.....	83
Introdução a JavaScript.....	84
O que é JavaScript?.....	84
História de JavaScript e outras linguagens web.....	84
Compatibilidade nos <i>browsers</i>	85
Editores para JavaScript.....	85
Como usar JavaScript em uma página.....	85
Forma #1 – aplicando diretamente à tag HTML.....	85
Forma #2 – dentro de uma <script> na página.....	85
Forma #3 – em um arquivo .js.....	85
Depurando scripts.....	86
Observações.....	86
Saiba mais.....	86
Document Object Model e Eventos.....	87
Modelo DOM.....	87
Principais objetos DOM.....	87
Objeto window.....	87
Objeto document.....	88
Objeto location.....	88
Objeto history.....	88
Objeto navigator.....	89
Eventos.....	89
Principais eventos.....	89

Principais eventos (mouse).....	89
Principais eventos (teclado).....	90
Objeto event.....	90
Tipos de Dados, Variáveis e Operadores.....	91
Tipos de dados.....	91
Funções para manipulação de strings e números.....	91
Variáveis.....	92
Declaração de variáveis.....	92
Operadores.....	92
Operadores aritméticos.....	92
Atribuições.....	92
Operadores relacionais.....	93
Operadores lógicos.....	93
Operadores para string.....	93
Operadores especiais.....	93
Funções e comandos de controle e de repetição.....	94
Criando funções.....	94
Exemplos de funções.....	94
Comandos de controle de fluxo.....	94
if ... else - Sintaxe.....	95
if ... else - Exemplo.....	95
switch - Sintaxe.....	95
switch - Exemplo.....	95
Comandos de repetição.....	96
while - Sintaxe.....	96
while - Exemplo.....	96
for - Sintaxe.....	96
for - Exemplo.....	96
for...in - Sintaxe.....	97
for...in - Exemplo.....	97
Comandos de interrupção.....	97
Objetos e Arrays.....	98
Objetos.....	98
Declarando objetos.....	98
Atribuindo valores às propriedades.....	98
Definindo métodos.....	99
Arrays.....	99
Declarando Arrays.....	99
Atribuindo valores ao Array.....	99
Propriedades e Métodos de um Array.....	100
Date, Math, String e Object.....	101
Date.....	101
Métodos de um objeto Date.....	101
Math.....	101
Propriedades e Métodos de Math.....	101
String.....	102
Propriedades e Métodos de um objeto String.....	102
Object.....	102
Propriedades e Métodos de Object.....	102

Parte 5 - Bootstrap.....	103
Introdução ao Bootstrap.....	104
O que é Bootstrap?.....	104
Exemplos de regras CSS presentes no Bootstrap.....	104
Como usar o Bootstrap?.....	104
Como usar em uma página HTML?.....	104
Estrutura básica de uma página.....	105
Estrutura de página com três seções.....	105
Ícones, Breakpoints e Sistema de Grid.....	107
Ícones.....	107
Breakpoints e visibilidade dos elementos.....	107
Renderizando imagens.....	108
Sistema de Grid.....	108
Exemplos de página com três colunas.....	109
Alguns Componentes no Bootstrap.....	110
Barra de navegação.....	110
Barra de navegação com Dropdown.....	111
Cards.....	112
Jumbotron.....	113
Badges.....	113
Alertas.....	114
Barra de ferramentas.....	114
Breadcrumb Navigation Bar.....	115
Paginação.....	115
Formatando tabelas e formulários.....	117
Formatando tabelas.....	117
Formatando formulários.....	117
Alguns componentes mais avançados.....	119
Janela modal.....	119
Carrossel.....	120
Precisa de Mais Componentes?.....	121
Criando uma galeria de imagens.....	122
Como criar uma galeria de imagens?.....	122

Parte 1 – HTML 5

Introdução à HTML 5

O que é HTML?

- HyperText Markup Language (Linguagem de Marcação de Hipertexto);
- Um documento HTML é um arquivo texto que emprega marcações (tags) para descrever o conteúdo de páginas web, que será interpretado e renderizado pelos navegadores.

Evolução da web

Pré-web

- Dados isolados (sem uma grande rede de comunicação e acesso aos dados);
- A Internet nasceu, mas faltava uma ferramenta para facilitar o compartilhamento de informações por meio dela;
- Surge a World Wide Web!

Desafio

- O que é a World Wide Web?
- Quem definiu/inventou a Web?

Web 1.0

- Conceito surgiu em 1990, por Tim Bernes Lee;
- Junção de três tecnologias simples: HTML, HTTP e Hyperlinks;
- Web meramente informativa (estática).

Web 2.0

- Termo surgiu em meados de 2004, por Dale Dougherty;
- Desenvolvimento de tecnologias (linguagens de programação) server-side;
- Todos passam a ser produtores e consumidores de informação;
 - Redes sociais, blogs e sites de compartilhamento de multimídia;
 - Crescimento exponencial do volume de dados.
- Web colaborativa (dinâmica).

Web 3.0

- Processar volumes enormes de dados exige que computadores compreendam suas informações para filtragem e processamento;
- Nesse cenário, encontramos a HTML 5, que permite uso de marcações para melhor definir o significado de conteúdos em páginas Web;
- Web semântica.

Web 4.0

- “Segundo Seth Godin, será como um gigantesco SO inteligente e dinâmico, que irá suportar as interações dos indivíduos, utilizando os dados disponíveis, instantâneos ou históricos, para propor ou suportar a tomada de decisão” (NETO, s.d.);
- Web inteligente;
- As tecnologias para tal experiência web ainda estão em desenvolvimento.

Desafio

- Para você, o que será a “web do futuro”?

Editores HTML

- Pode-se escrever páginas HTML em qualquer editor de texto...
 - Bloco de Notas, Notepad++ etc.
 - Lembre-se apenas de salvar com a extensão “.htm” ou “.html”!
- ...e ver o resultado produzido em qualquer navegador...
 - Google Chrome, Mozilla Firefox, Internet Explorer, Safari, Opera etc.
- Entretanto, há editores HTML que oferecem facilidades e recursos para o trabalho do desenvolvedor Web:
 - Adobe Dreamweaver (desktop pago)
<http://www.adobe.com/products/dreamweaver.html>
 - CoffeCup (desktop gratuito/pago)
<http://www.coffeecup.com/free-editor/>
 - Online HTML Editor (online gratuito)
<https://html-online.com/editor/>
 - Eclipse (desktop gratuito)

Estrutura Básica de um Documento HTML

Recordando...

- O que é HTML?
- O que é a Web 2.0? E a Web 3.0?
- O que significa semântica?

Estrutura básica de uma tag

```
<tag atributo="valor">  
    Conteúdo da tag  
</tag>
```

ou

```
<tag atributo="valor">
```

- Tags que apresentam conteúdo devem ser abertas (<tag>) e fechadas (</tag>);
- Algumas tags não apresentam conteúdo (ex:
). Nesses casos a tag é somente aberta;
- Atributos definem certas características de uma tag;
- A partir da HTML 5, todos os atributos antes usados para formatação de uma tag foram removidos. Agora a formatação da mesma será feita por meio de CSS 3!

Estrutura básica de um documento HTML

```
<!DOCTYPE html>  
<html>  
    <head>  
    </head>  
    <body>  
    </body>  
</html>
```

- Declaração DOCTYPE – define o tipo de documento;
- Tag html – envolve todo o conteúdo de uma página web;
- Tag head – define um “cabeçalho” (informações referentes à página em si);
- Tag body – apresenta o conteúdo a ser renderizado pelo navegador.
- Tags não são case-sensitive (sensíveis à caixa);
- Toda tag aberta deve ser fechada – exceto tags que não apresentam conteúdo!
- Tags mais internas devem ser fechadas antes das tags mais externas.

Exercício

Crie um documento HTML que, ao ser aberto no navegador, exiba a seguinte mensagem:

Olá, mundo!

Atributos globais mais comuns

Atributo	Descrição
class	Especifica um ou mais nomes de classes (da folha de estilo) para um elemento
id	Especifica um identificador único para um elemento
style	Especifica o estilo CSS "inline" para o elemento
title	Especifica informação extra sobre o elemento, visível ao passar o mouse sobre ele

Tag Head

- Responsável por prover informações sobre a página;
- Algumas tags que podem ser usadas aqui:
 - title – define um título para a página;
 - meta – provê metadados (dados sobre os dados), como tipo de codificação, palavras-chave, descrição etc.
 - link – estabelece uma relação entre o documento e um arquivo externo (estilo css, ícone etc.);
 - style – define regras de estilo (CSS) diretamente no documento;
 - script – provê instruções em linguagem de programação client-side (geralmente em JavaScript).

```
<head>
  <title>HTML Editors</title>
  <meta charset="utf-8">
  <meta name="keywords" content="HTML, CSS, JavaScript">
  <meta name="description" content="Well organized and easy to understand Web bulding tutorials with
lots of examples...">
  <link rel="icon" href="/favicon.ico" type="image/x-icon">
  <link rel="stylesheet" href="/lib/w3.css">
</head>
```

Extraído de: https://www.w3schools.com/html/html_editors.asp

Tag Body

- Descreve todo o conteúdo a ser renderizado pelo navegador;
- Algumas tags que podem ser usadas aqui:
 - Tags de cabeçalho: <h1>, <h2>... <h6>;
 - Tag de parágrafo: <p>;

- Tags para formatação: ou , <i> ou , <u> etc.
- Tags para hyperlink: <a>;
- Tags de imagem: ;
- Tags para listas: , e ;
- Tags para tabelas: <table>, <tr>, <th> e <td>;
- Tags para blocos: <div> e ;
- Tags para layout: <header>, <nav>, <section>, <article> etc.

```

<body>
  <h1>IFSBook – A rede social dos alunos do IFS</h1>
  <p>O IFSBook traz como proposta integrar as características de redes sociais como o Facebook, novas abordagens educacionais como aquelas propostas em ambientes virtuais de aprendizagem como o Khan Academy e um sistema que facilite o compartilhamento de monografias, artigos científicos e relatórios técnicos produzidos pelos estudantes do Instituto Federal de Sergipe.</p>
  <p>Assim, se você é estudante de nossa instituição, poderá acessar de forma bem simples todo o conteúdo escolhido especialmente para você.</p>
</body>

```

Exercício

Crie um documento HTML (biografia.html) com a sua “biografia”, isto é, o título da página (a aparecer na barra do navegador) deve ser o seu nome e no corpo da página escreva um parágrafo descrevendo a sua rotina diária.

Tags HTML Básicas

Recordando...

- Qual o papel da tag head?
- Qual o papel da tag body?
- Qual tag uso para especificar o título de minha página?

Tags Cabeçalhos (h1...h6)

- São usadas para identificar títulos de seção dentro de um documento HTML;
- Podem variar de h1 até h6, onde h1 é a mais importante (geralmente contendo o título da página) e h6 é a menos importante;
- Não confundir com a tag head!
 - h1...h6 – definem títulos e subtítulos de seções dentro da página, usadas dentro da tag body;
 - head – define informações sobre a página, como título da mesma, descrição, palavras-chave etc.
 - title – define o título da página que aparecerá na barra do navegador.

```
<h1>Lista de plantas silvestres na Amazônia</h1>
<p>A lista de plantas silvestres da Amazônia inclui todas as espécies endêmicas da vegetação da Amazônia. Segundo Giacometti (1990), é estimada a existência de 800 espécies vegetais de valor econômico ou social nesta floresta. Elas foram listadas abaixo de acordo com a família à qual pertencem.</p>
<h2>Bromeliaceae</h2>
<p>Bromeliaceae Juss. é uma família de monocotiledôneas, que segundo a classificação filogenética AGP III (2009) pertence à ordem dos Poales, com plantas terrestres, rupícolas ou, principalmente epífitas, possui 3.172 espécies, distribuídas em 58 gêneros.</p>
<h2>Chrysobalanaceae</h2>
<p>Chrysobalanaceae é uma família de plantas angiospérmicas (plantas com flor - divisão Magnoliophyta), pertencente à ordem Malpighiales.</p>
```

Tag Parágrafo (p)

- Cada parágrafo em um documento pode (e deveria) estar envolto por uma tag p;
- Caso uma tag p não seja fechada corretamente, a maioria dos navegadores o faz de forma implícita toda vez que encontra uma nova tag p aberta;
 - Afinal de contas, não dá para ter um parágrafo dentro de outro!
- Por padrão, os navegadores acrescentam algum espaço extra (margem) entre dois parágrafos consecutivos;
- Forma mais adequada de usar:
 - <p>Conteúdo aqui.</p>

Exercício

- Vamos agora atualizar o documento da última aula (biografia.html) da seguinte forma:
 - Acima do parágrafo com a biografia, deve aparecer um título para a página com o seu nome;
 - Abaixo do referido parágrafo, teremos uma nova seção chamada “Encontre-me”, na qual você deve colocar seu e-mail e nome no Facebook.

Tag Quebra de Linha (br)

- Ao inserir uma tag para quebra de linha (br) em uma página, “quebra-se” a linha e inicia uma nova, mas sem iniciar um novo parágrafo;
- Assim, a quebra de linha não acrescenta um espaço entre parágrafos, como ocorre com o uso da tag p;
- Como se trata somente de uma quebra de linha, não é possível aninhar outras tags dentro dela;
- Forma mais adequada de usar:


```
<h1>Lista de plantas silvestres na Amazônia</h1>
<p>A lista de plantas silvestres da Amazônia inclui todas as espécies endêmicas da vegetação da Amazônia. <br>
Segundo Giacometti (1990), é estimada a existência de 800 espécies vegetais de valor econômico ou social nesta
floresta. Elas foram listadas abaixo de acordo com a família à qual pertencem.</p>
<h2>Bromeliaceae</h2>
<p>Bromeliaceae Juss. é uma família de monocotiledôneas, que segundo a classificação filogenética AGP III
(2009) <br> pertence à ordem dos Poales, com plantas terrestres, rupícolas ou, principalmente epífitas, possui
3.172 espécies, distribuídas em 58 gêneros.</p>
<h2>Chrysobalanaceae</h2>
<p>Chrysobalanaceae é uma família de plantas angiospérmicas (plantas com flor - divisão Magnoliophyta),
pertencente à ordem Malpighiales.</p>
```

Tag Linha Horizontal (hr)

- Permite a inserção de uma linha horizontal na página;
- Pode ser usada para separar visualmente conteúdos em uma página;
- Forma mais adequada de usar:

<hr>

```
<h1>Lista de plantas silvestres na Amazônia</h1>
<p>A lista de plantas silvestres da Amazônia inclui todas as espécies endêmicas da vegetação da Amazônia. <br>
Segundo Giacometti (1990), é estimada a existência de 800 espécies vegetais de valor econômico ou social nesta
floresta. Elas foram listadas abaixo de acordo com a família à qual pertencem.</p>
<hr>
<p>Bromeliaceae Juss. é uma família de monocotiledôneas, que segundo a classificação filogenética AGP III
(2009) <br> pertence à ordem dos Poales, com plantas terrestres, rupícolas ou, principalmente epífitas, possui
3.172 espécies, distribuídas em 58 gêneros.</p>
<hr>
<p>Chrysobalanaceae é uma família de plantas angiospérmicas (plantas com flor - divisão Magnoliophyta),
pertencente à ordem Malpighiales.</p>
```

Tag Hyperlink (a)

- Usada para criar uma ligação (hyperlink) com outro documento ou arquivo disponível na web;
 - Também pode ser usada para disparar funções JavaScript!

- Ligação com outro documento:

`Ir para o Google`

- Ligação com outra parte do mesmo documento:

`Visite a seção de dicas`

...

`<h2 id="dicas">Seção de Dicas</h2>`

- Ligação com outra parte em outro documento:

`Ir para seção 3 do capítulo 2`

Principais atributos

- Atributo href:
 - Especifica a URL da página ou arquivo alvo após o clique;
 - Pode referir-se a outro documento, outra parte do documento atual ou outra parte de outro documento;
 - Mais importante de todos os atributos desta tag!
 - Exemplos:
 - `clique`
Abrirá a referida página
 - `clique`
Irá para a seção referida (elemento HTML com id="main")
 - `clique`
Abrirá a referida página na seção indicada
 - `enviar e-mail`
Abrirá o cliente de e-mail instalado com o destinatário já preenchido.
- Atributo target:
 - Especifica onde abrir o documento linkado;
 - Valores mais comuns:
 - `_blank` – abrirá em nova aba/janela;
 - `_parent` – abrirá no “frame pai” (se hyperlink estiver dentro de frame);
 - `_self` – abrirá na janela/frame atual;

- `_top` – abrirá ocupando toda a janela atual, mesmo se estiver em um frame;
- `nome_do_frame` – abrirá na janela/aba/frame que tiver aquele nome.
- Exemplo:
 - `visitar`
Abrirá a página em uma nova aba/janela.

Exercício

- Ainda no documento “biografia.html”, vamos incluir dois hyperlinks:
 - Um no endereço de e-mail para abri-lo em um cliente de e-mail;
 - Outro no nome do Facebook para abrir sua página no Facebook.

Tags para Multimídia

Recordando...

- Qual o nome da tag para:
 - Título de seção / cabeçalho?
 - Parágrafo?
 - Quebra de linha?
 - Hyperlink?

Imagem

Especifica as informações acerca de uma imagem a ser exibida no documento;

Formatos de imagem mais utilizados: JPG/JPEG, PNG, GIF e BMP.

Principais atributos

- Atributo src:
 - Especifica a URL da imagem a ser exibida;
 - O atributo mais importante para a tag imagem.
 - Exemplos:
 - ``
Especificando o caminho completo (absoluto) para a imagem
 - ``
Especificando o caminho relativo para a imagem
- Atributo alt:
 - Especifica um texto alternativo para a imagem;
 - Útil tanto para fins explicativos, de acessibilidade e otimização para motores de busca.
 - Exemplo:
 - ``
Texto alternativo descrevendo a logomarca do website
- Atributos width e height:
 - Especificam as dimensões em pixels da imagem (width=largura, height=altura);
 - Se não forem especificadas, a imagem será exibida com 100% de seu tamanho. Se o valor especificado for diferente do tamanho real da imagem, ela será redimensionada.
 - Exemplo:
 - ``

A imagem será redimensionada para 300x300

Áudio

- A tag áudio é empregada para a execução de arquivos de áudio em um navegador;
- Essa tag suporta três formatos de arquivos diferentes: MP3, WAV e OGG. Mas o formato MP3 é o mais amplamente suportado pelos navegadores!
- Exemplo:
 - `<audio src="http://www.w3schools.com/tags/horse.mp3" controls>`
 - Seu navegador não suporta a tag audio.
 - `</audio>`

Principais atributos

- `autoplay` – especifica que o áudio deve ser reproduzido assim que estiver pronto;
- `controls` – especifica que os controles do áudio deveriam ser exibidos;
- `loop` – especifica que o áudio deve ser executado “em loop”;
- `src` – especifica a URL do arquivo de áudio.

Vídeo

- De forma análoga à tag áudio, a tag vídeo é usada para a reprodução de vídeos;
- Essa tag suporta três formatos de arquivo diferentes: MP4, WebM, Ogg. Entretanto, MP4 é o formato mais amplamente suportado pelos navegadores;
- Exemplo:
 - `<video src="https://www.w3schools.com/tags/movie.mp4" controls>`
 - Seu navegador não suporta a tag vídeo.
 - `</video>`

Principais atributos

- `autoplay` – especifica que o vídeo deve ser reproduzido assim que estiver pronto;
- `controls` – especifica que os controles do vídeo deveriam ser exibidos;
- `height` – especifica a altura do vídeo em pixels;
- `loop` – especifica que o vídeo deve ser executado “em loop”;
- `src` – especifica a URL do arquivo de vídeo.
- `width` – especifica a largura do player de vídeo em pixels.

Exercício

- Atualize o arquivo “biografia.html” para conter as seguintes informações:
 - Uma foto sua, se não tiver:

- Homem: <https://christianosantos.com/files/cs1/avatar-homem.jpg>
- Mulher: <https://christianosantos.com/files/cs1/avatar-mulher.jpg>
- Um vídeo seu, se não tiver:
 - <https://www.w3schools.com/tags/movie.mp4>
- Uma música de fundo, em loop:
 - <https://christianosantos.com/files/cs1/musica01.mp3>
 - <https://christianosantos.com/files/cs1/musica02.mp3>
 - <https://christianosantos.com/files/cs1/musica03.mp3>

Tags para Formatação

Recordando...

- Como proceder para exibir/executar os seguintes arquivos em uma página web:
 - figura.jpg
 - musica.mp3
 - filme.mp4

Tags para Negrito

- É possível aplicar negrito, itálico ou sublinhado tanto por meio de tags quanto por meio de definição de CSS;
- Quanto às tags para negrito, pode-se utilizar a tag `` ou a tag ``;
 - `Texto` - texto é destacado em negrito, mas não é dada ênfase ao seu conteúdo;
 - `Texto` - texto é destacado em negrito e dada ênfase ao seu conteúdo (útil para motores de busca).

Exercício

- Abra o arquivo “biografia.html” e destaque em negrito as palavras mais importantes.

Tags para Itálico

- Quanto às tags para itálico, pode-se utilizar a tag `<i>` ou a tag ``;
 - `<i>Texto</i>` - texto é destacado em itálico, mas não é dada ênfase ao seu conteúdo;
 - `Texto` - texto é destacado em itálico e dada ênfase ao seu conteúdo (útil para motores de busca).

Exercício

- Ainda no arquivo “biografia.html”, inclua abaixo do parágrafo com sua descrição uma citação que goste muito (em itálico) e o nome do autor. Exemplo:

“Só sei que nada sei.” ~ Sócrates

Tags para Sublinhado

- Quanto à tag para sublinhado, pode-se utilizar a tag `<u>`;
 - `<u>Texto</u>` - texto é destacado com um sublinhado.
- Entretanto, não se recomenda seu uso para evitar confusão entre texto sublinhado e hyperlinks;
- Recomenda-se seu uso apenas para destacar textos estilisticamente diferentes do texto normal, como palavras escritas incorretamente (propositalmente!) ou nomes próprios em chinês;
 - Ex: `<p>A palavra estrambólico na verdade se escreve estrambótico.</p>`

Tag para Texto Pré-Formatado

- Ao usar a tag `p` ou mesmo escrevendo livremente no corpo do documento, o navegador ignorará quebras de linhas e espaços múltiplos presentes no texto.

Código HTML	No navegador
<pre><p> Era uma casa Muito engraçada Não tinha teto Não tinha nada </p></pre>	Era uma casa Muito engraçada Não tinha teto Não tinha nada

- Para manter a mesma formatação quanto a espaços e quebras de linha que apresente no código, pode-se usar a tag `pre`:

Código HTML	No navegador
<pre><pre> Era uma casa Muito engraçada Não tinha teto Não tinha nada </pre></pre>	Era uma casa Muito engraçada Não tinha teto Não tinha nada

Tags para subscrito e sobrescrito

- Caso precise representar letras ou números em subscrito (como em H_2O) ou sobrescrito (como em $30 m^3$), é possível fazer isso por meio das seguintes tags:
 - `_{Texto}` - O texto dentro da tag ficará subscrito.
 $H_{2}O$
 - `^{Texto}` - O texto dentro da tag ficará sobrescrito.
 $30 m^{3}$

Exercício

- Crie um novo documento HTML, contendo as seguintes frases:
 - A fórmula da água é H_2O , já o ozônio é O_3 .
 - Tenho uma casa de $90 m^2$.
 - 1000 litros de água ocupam $1 m^3$.

Tag para marcado, deletado e inserido

- Às vezes, queremos destacar com maior ênfase visual (tag `mark`) ou deixar claro que um trecho do texto foi removido (tag `del`) ou inserido (tag `ins`). Abaixo, exemplos de como usar tais tags:
 - HTML `<mark>Marked</mark>` Formatting

HTML **Marked** Formatting

- HTML `Deleted` Formatting

HTML ~~Deleted~~ Formatting

- HTML `<ins>Inserted</ins>` Formatting

HTML Inserted Formatting

Tabelas, Listas e Iframes

Recordando...

- Qual(is) tag(s) posso usar para:
 - Negrito?
 - Itálico?
 - Sublinhado?

Tags para Tabelas

- Podemos exibir dados em formato tabular (linhas e colunas). Para tal, precisamos conhecer as tags para criar uma tabela, suas linhas e células, que são:
 - <table> – define a tabela em si. É composta por um conjunto de linhas;
 - <tr> – define cada linha, isto é, conjunto de células dispostas horizontalmente;
 - <td> - define cada célula de uma linha, dentro da qual podemos colocar a informação desejada;
 - <th> - um “tipo especial” da tag td, usada para identificar as células que fazem parte do cabeçalho da tabela.
- Por padrão, uma tabela em HTML aparecerá sem as bordas. Aprenderemos mais tarde como definir suas bordas em CSS 3, mas por agora usaremos o antigo atributo border para defini-la, assim:

```
<table border="1">
```

```
<table border="1">
  <tr>
    <td>Aluno</td><td>Nota</td>
  </tr>
  <tr>
    <td>Carlos</td><td>10,0</td>
  </tr>
  <tr>
    <td>Maria</td><td>10,0</td>
  </tr>
</table>
```

Aluno	Nota
Carlos	10,0
Maria	10,0

```
<table border="1">
  <tr>
    <th>Aluno</th><th>Nota</th>
  </tr>
  <tr>
    <td>Carlos</td><td>10,0</td>
  </tr>
  <tr>
    <td>Maria</td><td>10,0</td>
  </tr>
</table>
```

Aluno	Nota
Carlos	10,0
Maria	10,0

Atributos rowspan e colspan

- Em certas situações, precisamos expandir uma célula para que a mesma ocupa mais de uma linha ou coluna. Nesses casos, podemos utilizar o atributo da célula (tag td) **colspan** para definir o número de colunas que ela ocupará ou o atributo **rowspan** para definir o número de linhas.

<pre><table border="1"> <tr> <th>Aluno</th><th>Nota</th> </tr> <tr> <td colspan="2">Carlos</td> </tr> <tr> <td>Maria</td><td>10,0</td> </tr> </table></pre>	<table border="1"><thead><tr><th>Aluno</th><th>Nota</th></tr></thead><tbody><tr><td colspan="2">Carlos</td></tr><tr><td>Maria</td><td>10,0</td></tr></tbody></table>	Aluno	Nota	Carlos		Maria	10,0
Aluno	Nota						
Carlos							
Maria	10,0						

Exemplo com colspan

<pre><table border="1"> <tr> <th>Aluno</th><th>Nota</th> </tr> <tr> <td rowspan="2">Carlos</td> <td>10,0</td> </tr> <tr> <td>10,0</td> </tr> </table></pre>	<table border="1"><thead><tr><th>Aluno</th><th>Nota</th></tr></thead><tbody><tr><td rowspan="2">Carlos</td><td>10,0</td></tr><tr><td>10,0</td></tr></tbody></table>	Aluno	Nota	Carlos	10,0	10,0
Aluno	Nota					
Carlos	10,0					
	10,0					

Exemplo com rowspan

- Além disso, é possível definir um rótulo a aparecer no topo da tabela, junto à mesma, especificando uma tag caption logo após abrir a tag table. Exemplo:

<pre><table border="1"> <caption>Resultado 1º Bimestre</caption> <tr><th>Aluno</th><th>Nota</th></tr> <tr><td>Carlos</td><td>10,0</td></tr> <tr><td>Maria</td><td>10,0</td></tr> </table></pre>	<p>Resultado 1º Bimestre</p> <table border="1"><thead><tr><th>Aluno</th><th>Nota</th></tr></thead><tbody><tr><td>Carlos</td><td>10,0</td></tr><tr><td>Maria</td><td>10,0</td></tr></tbody></table>	Aluno	Nota	Carlos	10,0	Maria	10,0
Aluno	Nota						
Carlos	10,0						
Maria	10,0						

Exercício

- Crie um documento HTML contendo as seguintes tabelas:

Resultados do 1º Bimestre

Resultados			
Aluno	Situação	Nota 1	Nota 2
Alan	Matriculado	9,0	8,0
Bruna	Matriculado	8,5	9,5
Carlos	Trancado		

Resultados do 2º Bimestre

		Português		Matemática	
Situação	Aluno	Nota 1	Nota 2	Nota 1	Nota 2
Matriculado	Alan	9,0	8,0	9,5	8,5
	Bruna	8,5	9,5	8,0	9,0
Trancado	Carlos				

Tags para Listas

- Se desejarmos exibir vários itens organizados em uma lista, podemos fazer isso por meio das tags ou e :
 - - Descreve uma lista não ordenada, isto é, que usará símbolos como marcadores e não números;
 - - Descreve uma lista ordenada, isto é, que usará números em ordem crescente para marcar cada item da lista;
 - - Esta tag representa cada um dos itens de uma lista e é a mesma para ambos os tipos de lista.

<p>Código HTML</p> <pre> Tomate; Arroz; Feijão. </pre>	<p>No navegador</p> <ul style="list-style-type: none"> • Tomate; • Arroz; • Feijão.
---	---

Exemplo de lista não ordenada

<p>Código HTML</p> <pre> Tomate; Arroz; Feijão. </pre>	<p>No navegador</p> <ol style="list-style-type: none"> 1. Tomate; 2. Arroz; 3. Feijão.
---	--

Exemplo de lista ordenada

Alterando os marcadores de listas ordenadas

- No caso de listas ordenadas, pode-se utilizar o atributo **type** da tag para definir como os itens da lista devem ser numerados. Type pode assumir os seguintes valores:
 - "1" – Serão usados números (comportamento padrão);

- “A” – Serão usadas letras maiúsculas;
- “a” – Serão usadas letras minúsculas;
- “I” – Serão usados algarismos romanos maiúsculos;
- “i” – Serão usados algarismos romanos minúsculos.

<p>Código HTML</p> <pre><ol type="A"> Tomate; Arroz; Feijão; </pre>	<p>No navegador</p> <p>A. Tomate; B. Arroz; C. Feijão;</p>
--	---

Exemplo de lista ordenada definindo tipo de marcador

Exercício

- Escreva o código HTML para a seguinte lista:

1. Plantas:

- a. Palma;
- b. Comigo-ninguém-pode;

2. Animais:

- i. Curió;
- ii. Sabiá;
- iii. Macaco-prego.

Tags para Iframes

- Um iframe é utilizado quando precisamos exibir o conteúdo de outra página web dentro de uma página web;
 - Por exemplo, para exibir a página do motor de busca do Google ou um vídeo do YouTube em uma área de meu website.
- Exemplos:
 - `<iframe src="http://www.google.com.br">Seu browser não suporta iframes</iframe>`
 - `<iframe width="560" height="315" src="https://www.youtube.com/embed/7CfiS_U9V9s" frameborder="0" allowfullscreen></iframe>`

Principais atributos

- src – especifica o endereço do documento a incorporar no mesmo;
- name – especifica um nome;
- height – especifica (em pixels) a sua altura;

- width – especifica (em pixels) a sua largura.

Exercício

- Crie um documento chamado “animais.html”, contendo:
 - Título da página: Animais em extinção;
 - Dois ou três parágrafos contendo informações sobre animais em extinção;
 - Destaque duas ou três palavras importantes do texto em negrito;
 - Uma lista não-ordenada contendo o nome de dez animais em extinção;
 - Um vídeo do YouTube sobre animais em extinção (você usará a tag iframe).
- Crie uma página chamada “manual-do-investidor.html”, contendo:
 - Título da página: Manual do Investidor;
 - Quatro hyperlinks que devem “abrir” em um iframe abaixo:
 - Gastar, poupar ou investir?
<https://christianosantos.com/files/cs1/manual-do-investidor/pagina-01.html>
 - Renda fixa ou variável?
<https://christianosantos.com/files/cs1/manual-do-investidor/pagina-02.html>
 - Conhecendo o seu perfil de investidor
<https://christianosantos.com/files/cs1/manual-do-investidor/pagina-03.html>
 - Planejamento financeiro
<https://christianosantos.com/files/cs1/manual-do-investidor/pagina-04.html>
 - Iframe (1000px de largura, 500px de altura) que exibirá inicialmente a página do primeiro hyperlink.

Tags para Formulários

Recordando...

- Quais as tags usadas para criar:
 - Uma tabela?
 - Uma lista não-ordenada?
 - Uma lista ordenada?
- Para que serve a tag iframe?

Tag Form

- Um formulário HTML é usado para receber entrada de dados do usuário;
- Exemplos de formulário:
 - Formulário para login;
 - Caixa de busca;
 - Página de contato;
 - Subscrição em newsletter;
 - Cadastro de produtos.
- Sintaxe básica de um formulário:

```
<form action="pagina-para-envio.jsp">
```

Aqui vão os elementos HTML que o usuário preencherá!

```
</form>
```

Exemplos de formulários

- Formulário para login

```
<form action="https://www.w3schools.com/action_page.php">
```

Nome do usuário: <input type="text" name="usuario">

Senha: <input type="password" name="senha">

<input type="submit" value="Login">

```
</form>
```

Nome do usuário:

Senha:

- Caixa de busca

```
<form action="https://www.w3schools.com/action_page.php">
```

```
<input type="text" name="busca" placeholder="Digite um filme">
```

```
<input type="submit" value="Procurar">
```

```
</form>
```



- Página de contato

```
<form action="https://www.w3schools.com/action_page.php">
```

```
Nome: <input type="text" name="nome"><br>
```

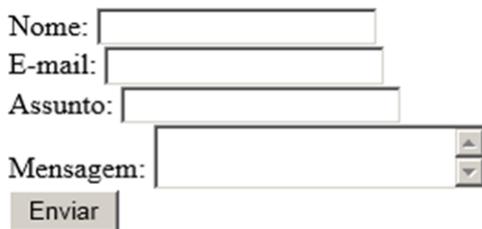
```
E-mail: <input type="email" name="email"><br>
```

```
Assunto: <input type="text" name="assunto"><br>
```

```
Mensagem: <textarea name="mensagem"></textarea><br>
```

```
<input type="submit" value="Enviar">
```

```
</form>
```



- Subscrição em newsletter

```
<form action="subscricao.jsp">
```

```
Nome: <input type="text" name="nome"><br>
```

```
E-mail: <input type="email" name="email"><br>
```

```
<input type="submit" value="Assinar">
```

```
</form>
```



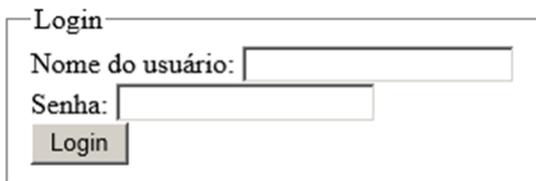
Tags fieldset e legend

- A tag **fieldset** pode ser usada para agrupar todos os elementos relacionados a um formulário;
 - Visualmente, ela criará uma borda em torno de todo o agrupamento.
- A tag **legend** pode ser usada dentro de uma tag fieldset para dar um rótulo ao agrupamento.

Tags fieldset e legend – exemplos

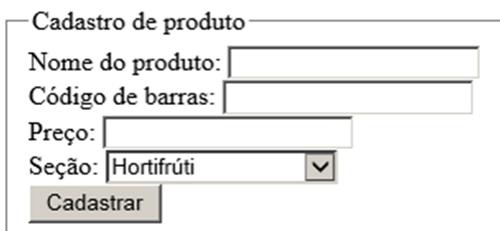
- Formulário de login

```
<form action="login.jsp">
  <fieldset>
    <legend>Login</legend>
    Nome do usuário: <input type="text" name="usuário"><br>
    Senha: <input type="password" name="senha"><br>
    <input type="submit" value="Login">
  </fieldset>
</form>
```



- Cadastro de produtos

```
<form action="cadastrar-produto.jsp">
  <fieldset>
    <legend>Cadastro de produto</legend>
    Nome do produto: <input type="text" name="nome"><br>
    Código de barras: <input type="text" name="codigo"><br>
    Preço: <input type="text" name="preco"><br>
    Seção: <select name="secao">
      <option value="hortifruti">Hortifrúti</option>
      <option value="carnes">Carnes</option>
    </select><br>
    <input type="submit" value="Cadastrar">
  </fieldset>
</form>
```



Principais atributos da tag form

- Action
 - Especifica o endereço URL onde submeter os dados do formulário.
 - Exemplo:

```
<form action="buscar.jsp">
    <input type="text" name="busca">
    <input type="submit" value="Procurar">
</form>
```
- Method
 - Especifica o método HTTP usado quando submetendo o formulário.
 - Valores: GET (valor padrão) ou POST.
 - Exemplo:

```
<form action="buscar.jsp" method="POST">
    <input type="text" name="busca">
    <input type="submit" value="Procurar">
</form>
```
- Target
 - Especifica a janela/frame alvo para abrir a página descrita em action.
 - Valores: `_blank`, `_parent`, `_self`, `_top`, `nome_do_frame`.
 - Exemplo:

```
<form action="buscar.jsp" target="_blank">
    <input type="text" name="busca">
    <input type="submit" value="Procurar">
</form>
```
- Enctype
 - Especifica o tipo de codificação dos dados submetidos.
 - Valores:
 - `application/x-www-form-urlencoded` (padrão);
 - `multipart/form-data`;
 - `text/plain`.

Exercício

- Crie três páginas HTML:
 - Uma página para login com os campos “nome de usuário” e “senha” (login.html);
 - Uma página para contato com os campos “nome”, “e-mail”, “assunto” e “mensagem” (contato.html);
 - Uma página para solicitação de filme com os campos “nome” e “ano” (solicitacao.html).

Tags Input, TextArea e Select

Recordando...

- Para que servem os seguintes atributos da tag form:
 - action
 - method
 - target
 - enctype

Tag Input

- Trata-se de uma tag para entrada (input) de dados do usuário;
- Uma das tags mais importantes e versáteis em um formulário. A depender de seu atributo type, ela poderá assumir vários papéis.

Alguns tipos de input

```
<input type="text" name="usuario" value="Mickey">
```

```
<input type="password" name="senha" value="Mouse">
```

```
<input type="date" name="dataEntrada">
```

dd/mm/aaaa → 01/mm/aaaa x ▾

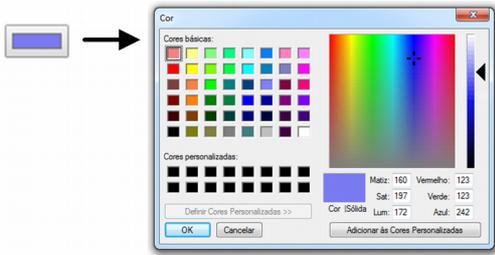
março de 2017 ◯ ◂ ◃

dom	seg	ter	qua	qui	sex	sáb
26	27	28	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

```
<input type="email" name="emailPrincipal" value="teste@gmail.com">
```

Obs: Uma vantagem de usar um input do tipo email é que, ao clicar no botão para submissão, o mesmo verifica automaticamente se o conteúdo corresponde a um endereço de e-mail válido.

```
<input type="color" name="corDeFundo">
```



```
<input type="radio" name="genero" value="homem" checked>Homem<br>
```

```
<input type="radio" name="genero" value="mulher">Mulher<br>
```

```
<input type="radio" name="genero" value="outro">Outro<br>
```

- Homem
- Mulher
- Outro

```
<input type="checkbox" name="preferencia" value="homem" >Homem<br>
```

```
<input type="checkbox" name="preferencia" value="mulher">Mulher<br>
```

```
<input type="checkbox" name="preferencia" value="outro">Outro<br>
```

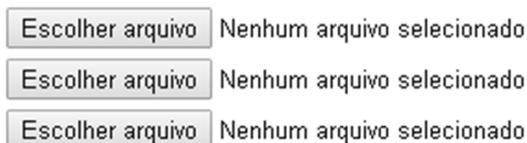
- Homem
- Mulher
- Outro

```
<input type="hidden" name="id" value="17">
```

```
<input type="file" name="arquivo">
```

```
<input type="file" name="arquivo" accept=".pdf">
```

```
<input type="file" name="arquivo" accept="image/*">
```



```
<input type="submit" name="btnEnviar" value="Enviar">
```



Outros atributos do input

- value – informação (valor) a ser exibida dentro do campo;
- maxlength – limita a quantidade máxima de caracteres que pode ser digitada;
- required – informa que o preenchimento do campo é obrigatório.

Outros tipos de input

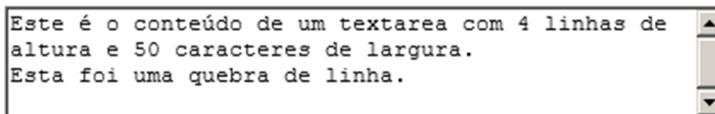
- Todos os valores possíveis para type: button, checkbox, color, date, datetime, datetime-local, email, file, hidden, image, month, number, password, radio, range, reset, search, submit, tel, text, time, url, week;
- Mais informações sobre tipos e atributos de input:
https://www.w3schools.com/tags/tag_input.asp

Tag TextArea

- Trata-se de uma tag para entrada de dados em formato de texto (aceita, inclusive, quebras de linha).
- Exemplo:

```
<textarea rows="4" cols="50">Este é o conteúdo de um textarea com 4 linhas de altura e 50 caracteres de largura.
```

```
Esta foi uma quebra de linha.</textarea>
```



Tag Select

- É usada para construir uma lista com várias opções, das quais o usuário poderá escolher uma ou mais;
- Cada elemento presente na lista será criado por meio de uma tag option.
- Exemplo de lista para selecionar um valor:

```
<select name="secao">
```

```
  <option value="hortifruti">Hortifrúti</option>
```

```
  <option value="carnes">Carnes</option>
```

```
  <option value="limpeza">Material de limpeza</option>
```

```
</select>
```



- Exemplo de lista para selecionar múltiplos valores (linhas):

```
<select name="secao" multiple>
```

```
  <option value="hortifruti">Hortifrúti</option>
```

```
  <option value="carnes">Carnes</option>
```

```
  <option value="limpeza">Material de limpeza</option>
```

```
</select>
```

Hortifrúti
Carnes
Material de limpeza

Exercício

- Crie uma página para cadastro de usuário (cadastro.html) contendo os seguintes campos:
 - Nome;
 - E-mail;
 - Senha;
 - Data de nascimento;
 - Sexo;
 - Estado.

Tag Input com Datalist

- Pode-se usar uma tag datalist associada a uma tag input para oferecer um conjunto de opções pré-definidas para o preenchimento;
- Nesse caso, o atributo "list" da tag input deve corresponder ao atributo "id" do datalist, a fim de fazer a associação entre eles.
- Exemplo:

```
<input list="browsers">
```

```
<datalist id="browsers">
```

```
<option value="Internet Explorer">
```

```
<option value="Firefox">
```

```
<option value="Chrome">
```

```
<option value="Opera">
```

```
<option value="Safari">
```

```
</datalist>
```



Outras Tags Úteis

Recordando...

- Quando usar a tag `textarea` em vez de um `input`?
- Cite três tipos da tag `input`;
- Para que serve a tag `select`?

Tag Div

- A tag **div** define uma seção em um documento, podendo conter quaisquer outras tags e/ou conteúdo textual;
- Pode ser formatada por meio de CSS, permitindo assim sua formatação e de seus elementos bem como o uso de tal tag para definição de layout de páginas web (tableless);
- Exemplo:

```
<div>
    <h3>Este é um cabeçalho</h3>
    <p>Este é um parágrafo.</p>
</div>
```

Tag Div – exemplo (a)

```
<div>
    <h3>Esportes</h3>
    <p>Notícia 1</p><p>Notícia 2</p><p>Notícia 3</p>
</div>
<div>
    <h3>Culinária</h3>
    <p>Notícia 1</p><p>Notícia 2</p><p>Notícia 3</p>
</div>
<div>
    <h3>Carros</h3>
    <p>Notícia 1</p><p>Notícia 2</p><p>Notícia 3</p>
</div>
```

Esportes

Notícia 1

Notícia 2

Notícia 3

Culinária

Notícia 1

Notícia 2

Notícia 3

Carros

Notícia 1

Notícia 2

Notícia 3

Tag Div – exemplo (b)

```
<div style="background-color:#FF0000;">
  <h3>Esportes</h3>
  <p>Notícia 1</p><p>Notícia 2</p><p>Notícia 3</p>
</div>
<div style="background-color:#00FF00;">
  <h3>Culinária</h3>
  <p>Notícia 1</p><p>Notícia 2</p><p>Notícia 3</p>
</div>
<div style="background-color:#0000FF;">
  <h3>Carros</h3>
  <p>Notícia 1</p><p>Notícia 2</p><p>Notícia 3</p>
</div>
```



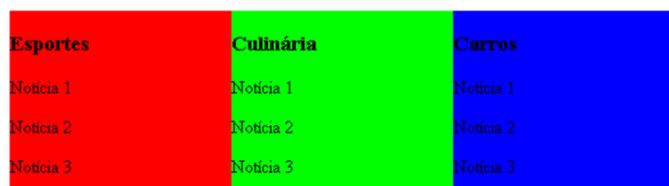
Tag Div – exemplo (c)

```
<div style="background-color:#FF0000;width:200px;">
  <h3>Esportes</h3>
  <p>Notícia 1</p><p>Notícia 2</p><p>Notícia 3</p>
</div>
<div style="background-color:#00FF00; width:200px;">
  <h3>Culinária</h3>
  <p>Notícia 1</p><p>Notícia 2</p><p>Notícia 3</p>
</div>
<div style="background-color:#0000FF; width:200px;">
  <h3>Carros</h3>
  <p>Notícia 1</p><p>Notícia 2</p><p>Notícia 3</p>
</div>
```



Tag Div – exemplo (d)

```
<div style="background-color:#FF0000;
width:200px;float:left;">
  <h3>Esportes</h3>
  <p>Notícia 1</p><p>Notícia 2</p><p>Notícia
3</p>
</div>
<div style="background-color:#00FF00; width:200px;
float:left;">
  <h3>Culinária</h3>
  <p>Notícia 1</p><p>Notícia 2</p><p>Notícia
3</p>
```



```
</div>
<div style="background-color:#0000FF; width:200px;
float:left;">
    <h3>Carros</h3>
    <p>Notícia 1</p><p>Notícia 2</p><p>Notícia
3</p>
</div>
```

Tag Span

- A tag **span** é usada para agrupar elementos inline (“em linha”) em um documento;
 - Exemplo: <p>Minha mãe tem olhos azuis.</p>
- Pode ser utilizada para aplicar estilo com CSS, incluir um identificador (a ser usado como “âncora interna” no documento) ou para manipulação via JavaScript.

Tag Label

- Pode-se usar a tag label para definir rótulos para os vários elementos HTML estudados nesta seção;
- Algumas vantagens de utilizar uma tag label em vez de simplesmente escrever antes do elemento:
 - Possibilidade de aplicar estilo CSS aos labels;
 - Possibilidade de acesso aos labels por meio de seu identificador;
 - Possibilidade de manipulação da tag label a partir de script.
- Exemplo

```
<label>Usuário</label> <input type="text" name="usuario">
```

Usuário

Tags Semânticas

- Tags como div e span, apesar de servirem para organizar o conteúdo em blocos ou “em linha”, não explicam muito do que se trata o conteúdo presente nas mesmas;
- Já tags como form, table e img conseguem definir claramente o tipo de conteúdo que carregam;
- A HTML5 introduz um novo conjunto de tags que, apesar de não serem obrigatórias, permitem uma melhor diferenciação quanto às diversas seções e conteúdos presentes na página.

Novas Tags Semânticas

<article>

<aside>

<details>

<figure>

<figcaption>

<footer>

<header>

<main>

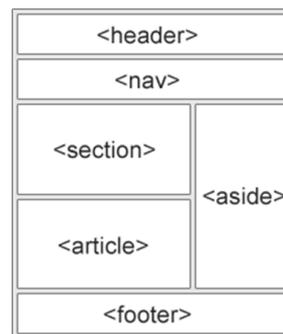
<mark>

<nav>

<section>

<summary>

<time>



Parte 2 – CSS 3

Introdução a CSS 3

O que é CSS?

- Acrônimo de Cascading Style Sheets (em português, “folhas de estilo em cascata”), uma “linguagem” de folhas de estilo usada para definir o layout de documentos HTML, XML e XHTML;
- Atualmente encontra-se na versão 3 (CSS 3);
- Graças a ela, podemos separar a definição das tags (HTML) da definição de formatação/layout (CSS):
 - O HTML fornece uma “estrutura” para o conteúdo;
 - O CSS traz toda a informação do layout (cores, posicionamento, fontes, tamanhos, imagens de fundo, margens etc.).

Benefícios da adoção de CSS¹

- Controle de interface em diferentes documentos em um único arquivo;
- Controle de diferentes interfaces para diferentes dispositivos (responsive design);
- Melhorias na acessibilidade com a possibilidade de “esconder” elementos da tela para usuários sem problemas de visão, mas manter os mesmos elementos acessíveis para leitores de tela;
- Menor consumo de banda para usuário e servidor;
- Inúmeras técnicas dinâmicas que não poderiam ser utilizadas em tabelas.

Editores CSS

- De forma análoga ao HTML, arquivos CSS são arquivos de texto, podendo ser criados ou editados em qualquer editor de textos (Bloco de Notas, Notepad++, Sublime Text etc.);
- Entretanto, alguns editores HTML oferecem mais funcionalidades durante a edição de arquivos CSS, como sintaxe colorida (destacando seletores, propriedades e valores) e autocompleção.

Como usar CSS em uma página

Forma #1 – aplicando diretamente na tag HTML

Exemplo:

```
<table style="width:300px;border:dashed;">
```

Simples para testar o efeito de uma regra CSS diretamente no HTML, entretanto pior escolha do ponto de vista de desenvolvimento e manutenção da página.

Forma #2 – dentro de uma <style> na <head>

Exemplo:

1 Fonte: <http://www.matera.com/br/2012/07/25/o-que-e-css-e-qual-sua-importancia/>

```

<style>

    table {

        width:300px;

        border:dashed;

    }

</style>

```

Já separa as regras CSS das tags HTML, mas ainda mantém ambos em um mesmo arquivo.

Forma #3 – em uma página .css

Exemplo:

estilo.css	index.html
<pre> table { width:300px; border:dashed; } </pre>	<pre> <!DOCTYPE html> <html> <head> ... <link rel="stylesheet" href="style.css"> ... </head> <body> ... </body> </html> </pre>

Aqui, temos separação completa entre HTML e CSS, o que permite o reúso de uma mesma folha de estilo em várias páginas.

Sintaxe CSS

- Cada arquivo .css (ou tag <style> em um documento HTML) define uma folha de estilo, que consiste em um conjunto de regras;
- Cada regra consiste de um ou mais seletores e um bloco de declaração;
- Cada bloco de declaração contém uma lista de declarações no formato “propriedade : valor ;”.

Folha de estilo	Exemplo (estilo.css)
<pre> seletor1 { propriedade: valor; } seletor2 { propriedade: valor; } seletor3:pseudoclasse { propriedade: valor; } </pre>	<pre> table { width: 300px; } td { background-color: RED; } a:hover { border: dashed; } </pre>

Seletores em CSS 3

Recordando...

- Qual o papel do CSS na construção de uma página HTML?
- Quais os três meios para usar estilo CSS em uma página HTML?

Seletores

- Um seletor é uma expressão usada para determinar o subconjunto de elementos HTML sobre os quais deve-se aplicar uma regra CSS (isto é, aplicar as propriedades associadas aos mesmos);
- Assim, um seletor pode assumir quatro formas básicas. Estudaremos agora as duas primeiras.

Tipos de Seletores

Forma #1 – Elementos do tipo

Especifica os elementos HTML (tags) sobre os quais se devem aplicar o efeito.

Sintaxe:

```
tag { ... }
```

Exemplo:

```
table { ... }
```

Aplicará o efeito descrito entre chaves a todas as tags do tipo table (tabela).

Forma #2 – Elementos da classe

Especifica a classe dos elementos sobre os quais se devem aplicar o efeito.

Sintaxe:

```
.classe { ... }
```

Exemplo:

```
.tblDados { ... }
```

Aplicará o efeito descrito entre chaves a todas as tags que apresentem a classe tblDados, exemplo: <table class="tblDados">

Obs: Pode-se aplicar mais de uma classe a uma tag separando-as por espaço.

Exemplo:

```
<table class="tblAnimal grid">
```

Nesse exemplo, essa tag table receberá as definições de estilo da classe tblAnimal e da classe grid (algo similar à herança múltipla de classes).

Forma #3 – Elementos com id

Especifica o id dos elementos sobre os quais se devem aplicar o efeito.

Sintaxe:

```
#id { ... }
```

Exemplo:

```
#tblAnimais { ... }
```

Aplicará o efeito descrito entre chaves a todas as tags que apresentem o id tblAnimais, exemplo: <table id="tblAnimais">

Obs: Apesar de ser possível ter mais de um elemento na mesma página com o mesmo id, na prática o ideal é que o id seja único para cada elemento. Assim sendo, caso deseje aplicar um efeito sobre mais de um elemento na mesma página é preferível usar uma classe em vez de adotar ids repetidos.

Forma #4 – Elementos com atributo

Especifica que o efeito deve ser aplicado somente a elementos que declarem aquele atributo.

Sintaxe:

```
[atributo] { ... }
```

Exemplo:

```
[target] { ... }
```

Aplicará o efeito descrito entre chaves a todas as tags que apresentem o atributo target, exemplo: Casa. **Obs:** Não será aplicada a <a>Casa.

Alguns operadores podem ser aplicados sobre o atributo. Um bastante comum é a igualdade.

Exemplo:

```
input[type="text"] { ... }
```

Aplicará o efeito descrito entre chaves a todas as tags input que apresentem o atributo type="text".

Para ver outros operadores, consulte: https://www.w3schools.com/css/css_attribute_selectors.asp

Pseudoclasses

- Uma pseudoclasa é usada para definir a formatação para um estado especial de um elemento. Por exemplo:
 - Para um elemento quando o cursor do mouse passa sobre ele (hover);
 - Para hyperlinks não visitados (link) ou já visitados (visited);
 - Para um elemento que recebeu o foco (focus).

Sintaxe básica	Exemplo
<pre>seletor:pseudoclasa { propriedade : valor; }</pre>	<pre>a:link { color: #FF0000; } a:visited { color : #00FF00;</pre>

	}
--	---

Nesse exemplo, a cor vermelha será aplicada a todos os hyperlinks (tag “a”) quando o link ainda não foi visitado e a cor verde será aplicada a todos os hyperlinks quando o link já foi visitado.

- Obs: No caso das pseudoclasses para links, quando presentes em um CSS, elas devem aparecer na seguinte ordem para surtir o efeito esperado:
 - a:link
 - a:visited
 - a:hover
 - a:active

Pseudoclasses mais usadas

Pseudoclasse	Exemplo	Descrição do exemplo
:link	a:link	Seleciona todos os links que ainda não foram visitados
:visited	a:visited	Seleciona todos os links já visitados
:hover	a:hover	Seleciona o link sob o cursor do mouse no momento
:active	a:active	Seleciona o link ativo
:focus	input:focus	Seleciona o input que tem o foco
:invalid	input:invalid	Seleciona todos os inputs com um valor inválido

Saiba mais em: https://www.w3schools.com/css/css_pseudo_classes.asp

Pseudoelementos

- Um pseudoelemento é usado para definir a formatação para uma parte especificada de um elemento, por exemplo:
 - Para a primeira letra (first-letter) de um elemento;
 - Para a primeira linha (first-line) de um elemento;
 - Para a porção do elemento selecionada pelo usuário (selection).

Sintaxe básica	Exemplo
<pre>seletor::pseudoelemento { propriedade : valor; }</pre>	<pre>p::first-letter { color: #FF0000; font-size: xx-large; } p::first-line { color : #0000FF; font-variant: small-caps; }</pre>

Nesse exemplo, para cada parágrafo a primeira letra aparecerá em cor vermelha e tamanho maior e o restante da primeira linha aparecerá em cor azul e em “caixa alta pequena”.

Pseudoelemento	Exemplo	Descrição do exemplo
::first-letter	p::first-letter	Seleciona a primeira letra de cada elemento p
::first-line	p::first-line	Seleciona a primeira linha de cada elemento p
::selection	p::selection	Seleciona a porção de um elemento p selecionada pelo usuário

Combinação de Seletores

- Diversos seletores podem ser combinados a fim de reaproveitar blocos de declarações idênticos ou para produzir filtros mais seletivos;
- A seguir, alguns exemplos...

<pre>p, h1, div { color: #FF0000; text-align: center; }</pre> <p><i>Em vez de criar três blocos (p, h1 e div) com as mesmas declarações, podemos agrupar todas elas, separadas por vírgula.</i></p>	<pre>div a { color: #333333; text-align: left; }</pre> <p><i>Se desejarmos aplicar um efeito somente às tags "a" dentro de tag "div", citamos as tags na ordem pai-filho, separadas por espaço.</i></p>
<pre>div.conteudo { color: #333333; text-align: left; }</pre> <p><i>Combinando tag e classe, podemos selecionar somente as tags div que apresentem a classe conteudo.</i></p>	<pre>a:hover { color: #AAAAAA; }</pre> <p><i>Combinando tag e pseudoclassee, podemos selecionar somente um dado estado de uma tag.</i></p>
<pre>div#conteudo { color: #333333; text-align: left; }</pre> <p><i>Combinando tag e id, podemos selecionar somente as tags div que apresentem o id conteudo.</i></p>	<pre>a[target] { color: #333333; text-align: left; }</pre> <p><i>Combinando tag e atributo, podemos selecionar somente as tags a que declarem o atributo target.</i></p>

Para aprender outras combinações de seletores: https://www.w3schools.com/css/css_combinators.asp

Exemplo: Uma página HTML, múltiplos estilos

Disponibilizado pela W3Schools em: https://www.w3schools.com/css/css_intro.asp

Propriedades para Background e Borda

Recordando...

- O que é um seletor?
- Quais as opções de seletores disponíveis?

Propriedades para background

- Usadas para definir um background (“aspecto de fundo”) para um elemento. Ao escolher um background (imagem ou colorido), certifique-se de que o mesmo não prejudica a leitura do texto ou visualização de outras imagens presentes na página!

Propriedade	Descrição
background	Configura todas as propriedades de background em uma declaração
background-attachment	Especifica se a imagem de background é fixa ou se deve “rolar” junto com a página (scroll)
background-color	Configura a cor de background de um elemento
background-image	Configura uma imagem de background de um elemento
background-position	Configura a posição inicial de uma imagem de background
background-repeat	Configura como a imagem de background será repetida
background-size	Configura o tamanho da imagem de background

Propriedade background-color

```
body {  
    background-color: #FFF;  
}
```

A cor pode ser especificada por meio de uma das constantes (red, green, blue, orange, yellow, cyan, black, white etc.) ou por meio do valor RGB (ex: #FF0000 ou #F00 para vermelho).

Você pode usar o seguinte website para, a partir de uma cor selecionada, obter seu valor RGB: <http://www.colorpicker.com>

Propriedade background-image

```
body {  
    background-image: url("logo.jpg");  
}
```

O caminho especificado pode ser um caminho absoluto ou relativo.

Caso especifique tanto background-color quanto background-image, a cor preencherá todo o fundo enquanto a imagem for carregada e esta aparecerá sobre aquela após o carregamento.

Propriedade background-size

```
body {  
    background-image: url("background.jpg");  
    background-size: 100% 100%;  
}
```

Possíveis valores para o tamanho da imagem de background:

- auto – valor padrão. Mantém as dimensões da imagem original;
- 75px / 75% – define a largura/altura em pixels ou em percentual da área do elemento pai;
- cover – escala a imagem para ser tão grande quanto possível para cobrir toda a área de fundo;
- contain – escala a imagem para o maior tamanho possível tal que sua largura e altura ainda encaixem na área de fundo;
- initial – configura para o valor padrão do navegador;
- inherit – herda o valor do seu elemento pai.

Propriedade background-repeat

```
body {  
    background-image: url("background.jpg");  
    background-repeat: repeat-x;  
}
```

Possíveis valores para a repetição da imagem de background:

- repeat – repete-a horizontal e verticalmente;
- repeat-x – repete-a somente horizontalmente;
- repeat-y – repete-a somente verticalmente;
- no-repeat – não será repetida;
- initial ou inherit.

Propriedade background-position

```
body {  
    background-image: url("background.jpg");  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```

Os valores possíveis são:

- Para X (primeiro parâmetro) – left, center, right, 75%, 75px;
- Para Y (segundo parâmetro) – top, center, bottom, 75%, 75px.

Propriedade background-attachment

```
body {
    background-image: url("background.jpg");
    background-repeat: no-repeat;
    background-position: right top;
    background-attachment: fixed;
}
```

Os valores possíveis são: scroll (padrão), fixed, local, initial, inherit.

Propriedade background

```
body {
    background: #FFF url("img_tree.png") no-repeat right top;
}
```

Ao usar esta propriedade, a ordem das propriedades configuradas é:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

Propriedades para borda

- Usadas para definir bordas em um elemento (ex: as células de uma tabela);
- Abaixo, algumas propriedades para especificação de borda (veja todas em: https://www.w3schools.com/css/css_border.asp):

Propriedade	Descrição
border	Configura todas as propriedades das bordas em uma só declaração
border-color	Configura a cor para as bordas
border-radius	Configura o raio para bordas com cantos arredondados
border-style	Configura o estilo para as bordas
border-width	Configura a largura (espessura) das bordas

Propriedade border-style

```
td {  
    border-style: dashed;  
}
```

Os valores possíveis são: dotted, dashed, solid, double, groove, ridge, inset, outset, none, hidden.

Propriedade border-width

```
td {  
    border-style: solid;  
    border-width: 1px;  
}
```

Propriedade border-color

```
td {  
    border-style: solid;  
    border-width: 1px;  
    border-color: red;  
}
```

Propriedade border-radius

```
td {  
    border-style: solid;  
    border-width: 1px;  
    border-radius: 5px;  
}
```

Propriedade border

```
td {  
    border: 2px solid red;  
}
```

Ao usar esta propriedade, a ordem das propriedades estabelecidas é:

- border-width
- border-style
- border-color

Exercício

- Crie uma página HTML que contenha uma imagem de fundo que se ajuste para ocupar todo o fundo completamente;
- Nessa mesma página, inclua uma tag H1 com o conteúdo “Primeiro exercício de CSS” e faça com que essa tag tenha uma cor de fundo vermelha e uma borda verde de 2 pixels de espessura.

Propriedades para contorno

- As propriedades *outline* (contorno) são usadas para definir uma linha de contorno que ficará ao redor da margem do elemento;
- Obs: não confundir com a borda, que fica em torno do próprio elemento, entre a margem e o padding!

Propriedade	Descrição
outline	Configura todas as propriedades do contorno em uma só declaração
outline-color	Configura a cor para o contorno
outline-offset	Configura o espaço entre o contorno e a borda do elemento
outline-style	Configura o estilo para o contorno
outline-width	Configura a largura (espessura) do contorno

Propriedades outline-color, outline-offset, outline-style, outline-width

```
div {
  outline-color: #FFFF00;
  outline-offset: 5px;
  outline-style: dashed;
  outline-width: 3px;
}
```

Propriedade outline

```
div {
  outline: 3px dashed red;
}
```

De forma análoga à propriedade border, as propriedades configuradas são outline-width, outline-style e outline-color.

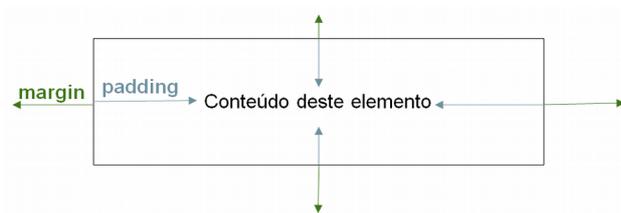
Propriedades para margem, padding, largura e altura

Recordando...

- Que propriedades usar para um DIV ter cantos arredondados e cor de fundo verde?
- Que propriedade permite definir a forma como a imagem de fundo deve se mover ao rolar a página?

Propriedades para margem e padding

- Tanto *margin* quanto *padding* servem para dar um espaçamento nas quatro direções:
 - *margin* – cria um espaçamento entre o elemento e o que está FORA dele (elementos em torno dele);
 - *padding* – cria um espaçamento entre o elemento e o que está DENTRO dele (elementos que compõem seu conteúdo).



Propriedades para margem

Propriedade	Descrição
margin	Configura todas as margens
margin-top	Configura a margem do topo
margin-right	Configura a margem da direita
margin-bottom	Configura a margem de baixo
margin-left	Configura a margem da esquerda

Propriedades *margin-top*, *margin-right*, *margin-bottom*, *margin-left*

```
div {  
    margin-top: 10px;  
    margin-right: 10px;  
    margin-bottom: 10px;  
    margin-left: 10px;  
}
```

Exemplos de valores possíveis são: auto, 75px, 75%, inherit.

Propriedade *margin*

```
td {  
    margin: 10px 20px 30px 40px;
```

```
}
```

Ao usar esta propriedade, a ordem das propriedades estabelecidas é (sentido horário):

- margin-top
- margin-right
- margin-bottom
- margin-left

Propriedades para *padding*

Propriedade	Descrição
padding	Configura todos os “padding”
padding-top	Configura o “padding” do topo
padding-right	Configura o “padding” da direita
padding-bottom	Configura o “padding” de baixo
padding-left	Configura o “padding” da esquerda

Propriedades *padding-top*, *padding-right*, *padding-bottom*, *padding-left*

```
div {  
    padding-top: 10px;  
    padding-right: 10px;  
    padding-bottom: 10px;  
    padding-left: 10px;  
}
```

Exemplos de valores possíveis são: auto, 75px, 75%, inherit.

Propriedade *padding*

```
td {  
    padding: 10px 20px 30px 40px;  
}
```

Ao usar esta propriedade, a ordem das propriedades estabelecidas é:

- padding-top
- padding-right
- padding-bottom
- padding-left

Propriedades para largura e altura

- Usadas para definir as larguras (width) e alturas (height) atuais, mínimas e/ou máximas de um elemento;
- Definem as dimensões da área INTERNA do elemento, desconsiderando padding, borda e margem!

Propriedade	Descrição
width	Configura a largura de um elemento
min-width	Configura a largura mínima de um elemento
max-width	Configura a largura máxima de um elemento
height	Configura a altura de um elemento
min-height	Configura a altura mínima de um elemento
max-height	Configura a altura máxima de um elemento

Propriedades width, min-width, max-width

```
div {  
    width: 100%;  
    min-width: 150px;  
    max-width: 300px;  
}
```

Exemplos de valores possíveis são: auto, 75px, 75%, inherit.

Propriedades height, min-height, max-height

```
div {  
    height: 100%;  
    min-height : 150px;  
    max-height : 300px;  
}
```

Exemplos de valores possíveis são: auto, 75px, 75%, inherit.

O “Modelo Caixa” em CSS

- Cada elemento HTML pode ser considerado como uma “caixa”, isto é, um retângulo na página com seu conteúdo internamente;
- O “Modelo Caixa” em CSS (Box Model) refere-se à forma como o mesmo organiza as propriedades de um elemento para fins de formatação;
- Observe a imagem:

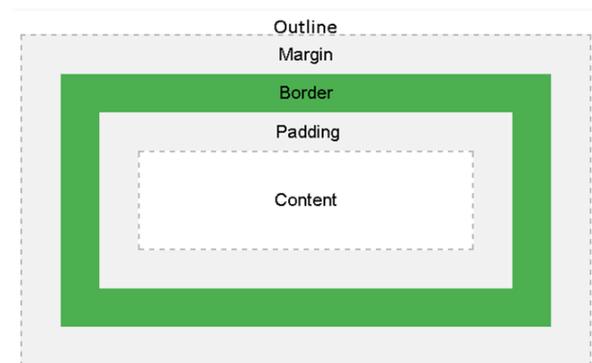


Largura e altura total de um elemento

- As propriedades CSS **width** e **height** de um elemento referem-se na verdade à área do conteúdo do elemento, e não ao elemento como um todo;
- Largura e altura total de um elemento podem ser determinadas somando-se as larguras e alturas do conteúdo, do padding, da borda e da margem.

Largura total = width +
 (padding-left + padding-right) +
 (border-left-width + border-right-width) +
 (margin-left + margin-right);

Altura total = height +
 (padding-top + padding-bottom) +
 (border-top-width + border-bottom-width) +
 (margin-top + margin-bottom);



Obs: A largura e altura (espessura) do contorno (outline) não é levada em consideração no cálculo da largura e altura do elemento uma vez que a mesma é “desenhada” em torno do mesmo, não interferindo em sua área.

Exercício

- Crie uma página com um DIV com fundo vermelho, 700px de largura, padding 0px e centralizado na página;
 - E agora, como centralizar um bloco?
- Agora insira um outro DIV, com fundo verde, 700 px de largura, padding 25px e também centralizado na página;
- Por que os dois DIV não estão completamente alinhados?

Propriedades para textos, fontes e listas

Recordando...

- Qual a diferença entre margin e padding?
- Como calcular a largura/altura total que um elemento ocupa na página?

Propriedades para textos

- Usadas para definir cor, alinhamento, direção e outras características relativas ao texto em si;
- Abaixo, algumas das propriedades mais usadas:

Propriedade	Descrição
color	Especifica a cor do texto
line-height	Especifica a altura da linha
text-align	Especifica o alinhamento horizontal do texto
vertical-align	Especifica o alinhamento vertical do texto
text-indent	Especifica a indentação da primeira linha de um bloco de texto
text-decoration	Especifica a “decoração” a ser adicionada ao texto

Propriedades color e line-height

```
p {  
  color: #555;  
  line-height: 1.5;  
}
```

Propriedades text-align e vertical-align

```
p {  
  text-align: center;  
  vertical-align: top;  
}
```

Possíveis valores para:

- text-align: left, right, center, justify, initial, inherit;
- vertical-align: baseline, 75px, 75%, sub, super, top, text-top, middle, bottom, text-bottom, initial, inherit.

Propriedades text-indent e text-decoration

```
p {  
  text-indent: 25px;
```

```
text-decoration: underline;
}
```

Possíveis valores para:

- text-indent: 75px, 75%, initial, inherit;
- text-decoration: none, underline, overline, line-through, initial, inherit;

Propriedades para fontes

- Usadas para definir o tipo de fonte (família), negrito, tamanho e estilo do texto.

Propriedade	Descrição
font	Especifica as propriedades de fonte em uma só declaração
font-family	Especifica a família da fonte para o texto
font-size	Especifica o tamanho da fonte para o texto
font-style	Especifica o estilo da fonte para o texto
font-variant	Especifica se o texto deveria aparecer em “caixa alta pequena” ou não
font-weight	Especifica a “densidade” (nível de negrito) do texto

Propriedade font-family

```
p {
  font-family: "Times New Roman", Georgia, Serif;
}
```

Possíveis valores para:

- Nome de uma família: “Times New Roman”, “Arial”, “Georgia” etc.
- Nome genérico: Serif, Sans-Serif, Cursive, Fantasy, Monospace.

Propriedade font-size

```
p {
  font-size: 20px;
}
```

Possíveis valores:

- xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger;
- 75px, 75%, 0.75em, 75pt, 0.75rem, 0.75cm, 0.75mm;
- initial, inherit.

Propriedade font-style

```
p {  
    font-style: italic;  
}
```

Possíveis valores: normal, italic, oblique, initial, inherit.

Propriedade font-variant

```
p {  
    font-variant: small-caps;  
}
```

Possíveis valores: normal, small-caps, initial, inherit.

Propriedade font-weight

```
p {  
    font-weight: bold;  
}
```

Possíveis valores:

- normal, bold, lighter, bolder, 100... 900, initial, inherit.

Propriedade font

```
p {  
    font: italic bold 12px/30px Georgia, serif;  
}
```

Possíveis valores:

- font-style font-variant font-weight font-size/line-height font-family
ou
- caption, icon, menu, message-box, small-caption, status-bar, initial, inherit.

Propriedades para listas

- Aplicáveis às listas ordenadas (ol) ou não-ordenadas (ul) e a seus itens.

Propriedade	Descrição
list-style	Especifica as propriedades para uma lista em uma só declaração
list-style-type	Especifica o tipo de marcador de item de lista
list-style-image	Especifica uma imagem como marcador de item de lista
list-style-position	Especifica se os marcadores de item de lista deveriam

Propriedade *list-style-type*

```
p {
  list-style-type: circle;
}
```

Alguns possíveis valores: disc, circle, square etc.

Veja a lista completa em: http://www.w3schools.com/cssref/pr_list-style-type.asp

Propriedade *list-style-image*

```
p {
  list-style-image: url("marcador.png");
}
```

Propriedade *list-style-position*

```
p {
  list-style-position: inside;
}
```

Possíveis valores: outside, inside.

Outside:

- Coffee
- Tea
- Coca-cola

Inside:

- Coffee
- Tea
- Coca-cola

Propriedade *list-style*

```
p {
  list-style: square url("square_purple.png");
}
```

Ordem das propriedades especificadas:

list-style-type list-style-position list-style-image.

Exercício

- Crie uma página HTML com um parágrafo qualquer contendo dois hyperlinks;
- Faça com que o todo o texto dentro do parágrafo seja sublinhado e use a fonte Arial, **exceto** os hyperlinks, que não devem ser sublinhados e sim aparecerem em negrito e com a fonte Times New Roman.

Propriedades para layout e transparência

Recordando...

- Que propriedade usar para:
 - Definir se o texto deve ser sublinhado;
 - Definir o tipo de fonte a ser usado;
 - Especificar o nível de negrito.

Propriedades para layout

- Usadas para melhor organizar o layout (visibilidade, posição etc.).

Propriedade	Descrição
display	Especifica como um elemento deveria ser mostrado (como bloco, “em linha” ou oculto)
float	Especifica se um elemento deveria “flutuar” ou não
clear	Especifica que o elemento deve ignorar a informação de “float” do elemento anterior, posicionando-o abaixo.

Propriedade display

```
span {  
    display: none;  
}
```

Alguns possíveis valores: inline, block, inline-block, none etc.

Propriedade float

```
div {  
    float: left;  
}
```

Possíveis valores: none, left, right, initial, inherit.

Propriedade clear

```
div {  
    clear: left;  
}
```

Possíveis valores: none, left, right, both, initial, inherit.

Propriedades para transparência

- Pode-se definir nível de transparência de um elemento HTML, criando efeitos (por exemplo, em galerias de imagens);

Propriedade	Descrição
opacity	Especifica quão opaco (não transparente) deve ser um elemento, varia de 0 (totalmente transparente) a 1 (totalmente opaco)
background-color: rgba(r, g, b, a);	Especifica uma cor transparente ao elemento HTML, onde r, g e b indicam os níveis de vermelho, verde e azul e a especifica a opacidade e varia de 0 (totalmente transparente) a 1 (totalmente opaco)

Propriedade opacity

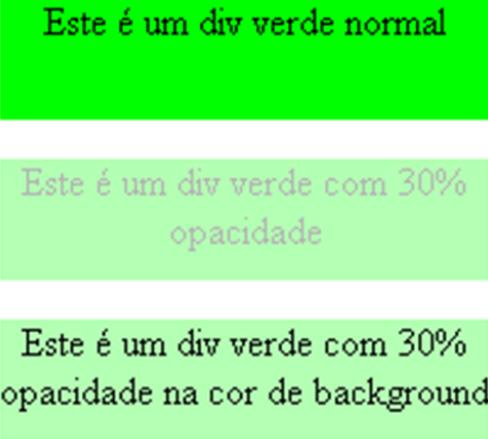
```
div {
  opacity: 0.5;
}
```

Ao definir a transparência de um elemento, ele e todos os seus elementos-filhos receberão aquele nível de transparência.

Propriedade background-color aplicando rgba

```
div {
  background-color: rgba(0, 255, 0, 0.5);
}
```

Ao definir uma cor de background transparente, alteramos a transparência somente do elemento atual, sem afetar seus elementos-filhos.

	<pre>.divNormal { background-color:#0F0; width:200px; height:50px; text-align:center; } .divOpacity { background-color:#0F0; width:200px; height:50px; text-align:center; opacity:0.3; } .divRGBA { background-color:rgba(0,255,0,0.3); width:200px; height:50px; text-align:center; }</pre>
---	--

Outras propriedades para layout

- Usadas para melhor organizar o layout (visibilidade, posição etc.).

Propriedade	Descrição
visibility	Especifica se um elemento deveria ser visível ou não
position	Especifica o tipo do posicionamento para o elemento
z-index	Especifica a ordem de exibição/desenho em tela

Propriedade visibility

```
span {  
    visibility: hidden;  
}
```

Possíveis valores: visible, hidden, collapse, initial, inherit.

Teste e responda: qual a diferença entre “display:none;” e “visibility:hidden;” ?

Propriedade position

```
div {  
    position: fixed;  
    right: 0px;  
    bottom: 0px;  
}
```

Possíveis valores para position: static, absolute, fixed, relative, initial, inherit.

Se a posição do elemento não é static (valor padrão), deve-se especificar valores para algumas das laterais (top, left, bottom e/ou right).

Propriedade z-index

```
div {  
    z-index: 2;  
}
```

Caso mais de um elemento se sobreponha na página (possível quando se altera a propriedade position), a propriedade z-index determinará qual elemento deve ser desenhado sobre o outro (quem possuir maior valor aparecerá “por cima”).

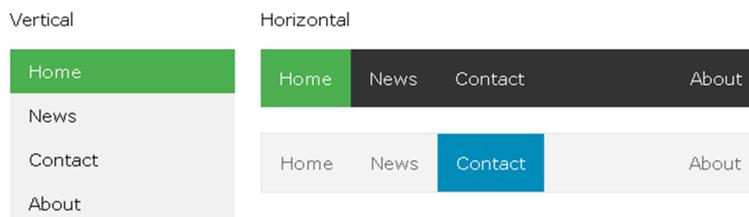
Exercício

- Crie uma página para uma loja de calçados contendo cinco DIVs com fundo verde (green), cada um exibindo uma imagem de calçado, bem como seu nome e preço. Os DIVs devem ficar lado a lado;
- Na mesma página, abaixo dos DIVs com os produtos deve aparecer um rodapé com cor de fundo cinza (gray) com o nome da loja;

- Crie uma página HTML longa (use qualquer texto) e inclua na mesma uma DIV com 200px de largura por 400px de altura com uma borda de 2px de espessura a permanecer sobre a página e fixa no canto inferior direito contendo o texto “Olá, nosso serviço de atendimento online encontra-se indisponível, tente novamente mais tarde”.

Barras de navegação em CSS

- Todo website apresenta uma barra de navegação, contendo os principais links para as áreas/funcionalidades disponíveis ao usuário!
- Uma barra de navegação com layout tanto horizontal quanto vertical pode ser implementada por meio de uma lista de links devidamente estilizada.



Criando uma barra de navegação (passo-a-passo)

1. [HTML] Crie sua barra de navegação como uma lista não ordenada de links e aplique uma classe active ao link atual:

```
<ul class="menulist ">  
  <li class="menuitem"><a href="#home" class="active" >Home</a></li>  
  <li class="menuitem"><a href="#news">News</a></li>  
  <li class="menuitem"><a href="#contact">Contact</a></li>  
  <li class="menuitem"><a href="#about">About</a></li>  
</ul>
```

2. [CSS] Para a lista, retire os marcadores, as margens e o espaçamento interno e defina a cor de background, a largura (se menu vertical) e overflow (se menu horizontal):

```
.menulist {  
  list-style-type: none;  
  margin: 0px;  
  padding: 0px;  
  background-color: #EEE;  
  overflow: hidden;  
}
```

3. [CSS] No caso de menu horizontal, para os itens de lista, defina float como left ou right:

```
.menuitem {  
  float:left;  
}
```

4. [CSS] Para link dentro de item de lista, defina display como block, text-decoration como none e a aparência:

```
.menuitem a {  
    display: block;  
    text-decoration: none;  
    color: #000;  
    padding: 10px;  
}
```

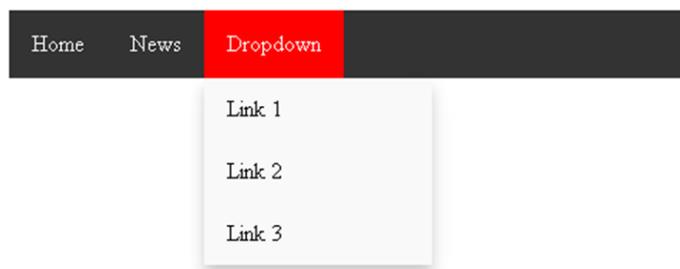
5. [CSS] Quando passar o mouse sobre o link dentro de item de lista, altere sua cor de background (background-color) e outros detalhes de aparência (color, por exemplo):

```
.menuitem a:hover {  
    background-color: #555;  
    color: #FFF;  
}
```

6. [CSS] Defina a aparência para o item ativo (classe active):

```
.active {  
    background-color: #4A5;  
    color: #FFF;  
}
```

Barra de navegação com menu dropdown (passo-a-passo)



0. Comece construindo sua barra de navegação horizontal (HTML e CSS), exatamente como na seção anterior;

1. [HTML] Acrescente a classe “dropbutton” ao hyperlink da opção que terá subopções (menu dropdown) e crie um div (classe “dropcontent”) com hyperlinks para representar o menu com as subopções:

```
<ul class="menulist">  
    <li class="menuitem"><a href="#home" class="active" >Home</a></li>  
    <li class="menuitem"><a href="#news">News</a></li>  
    <li class="menuitem">
```

```
<a href="#" class="dropbutton">Dropdown</a>
<div class="dropcontent">
  <a href="#">Link 1</a>
  <a href="#">Link 2</a>
</div>
</li>
</ul>
```

2. [CSS] Altere o seguinte seletor:

```
.menuitem a:hover, .menuitem:hover .dropbutton {
    background-color: #555555;
    color: #FFFFFF;
}
```

3. [CSS] Configure as propriedades para a classe "dropcontent" (div com as subopções) como sendo display:none (invisível inicialmente), position:absolute (aparecerá por cima dos outros elementos) e demais configurações de aparência:

```
.dropcontent {
    display: none;
    position: absolute;
    background-color: #EEE;
    min-width: 160px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
}
```

4. [CSS] Definir as opções de aparência para hyperlinks dentro de dropcontent:

```
.dropcontent a {
    color: black;
    padding: 12px 16px;
}
```

5. [CSS] Ao passar o cursor do mouse sobre um item de menu, o div com as subopções (classe "dropcontent") deve aparecer (display:block):

```
.menuitem:hover .dropdown-content {
    display: block;
}
```

6. [CSS] E finalmente, ao passar o cursor do mouse sobre uma das subopções, mudar a cor de seu background:

```
.dropdown-content a:hover {  
  background-color: #EEE;  
}
```

Exemplos

Barra de navegação vertical:

http://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_vertical_active

Barra de navegação horizontal:

http://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_horizontal_black_active

Barra de navegação horizontal com menu dropdown:

http://www.w3schools.com/css/tryit.asp?filename=trycss_dropdown_navbar

Parte 3 – Boas Práticas para a Web

Otimização para motores de busca

Otimização para motores de busca (SEO)

- Do inglês, *Search Engine Optimization* (SEO);
- Conforme motores de busca conquistaram maior relevância na web, tornou-se importante para websites de todos os tipos (lojas virtuais, blogs, portais de notícias etc.) conquistarem boas posições segundo as classificações dos mesmos;
- Trata-se, então, de um conjunto de boas práticas a serem adotadas por websites para tornar mais fácil a tarefa dos motores de busca de analisá-los e classificá-los segundo seu conteúdo.

Palavra-chave (por relevância)	Média de pesquisas mensais [?]	Concorrência [?]	Lance sugerido [?]
tenis infantil	10 mil – 100 mil	Alta	R\$0,92

- Cada motor de busca segue algoritmos próprios para classificar cada página na web, mas como o Google Search é o mais usado (70% - 80%), muitos desenvolvedores web focam sua atenção no mesmo;
- Mas a maioria das orientações (se não todas) servem para todos os motores de busca!

Fatores on-page e off-page a considerar

- São divididas em:
 - *On-page*: orientações focadas em detalhes da própria página que se deseja classificar (URL, título, qualidade do conteúdo, uso de multimídia, tempo de carregamento etc.);
 - *Off-page*: orientações focadas em análise de outras páginas, tanto aquelas presentes no mesmo website (estrutura hierárquica do website, links internos etc.) quanto aquelas presentes em outros websites (menções e citações à sua página – backlinks);
- Para alguns marqueteiros, SEO resume-se em palavras-chave, conteúdo e backlinks!

Guia para iniciantes (da Google)²

- Títulos únicos e precisos para cada página (máximo de 60 caracteres e contendo a palavra-chave ou variação da mesma);
- Meta tags description únicas para cada página contendo resumo em 140 caracteres;
- Cada página deve conter uma URL não muito longa, apresentando a palavra-chave ou variação. Exemplos:
 - <http://dominio.com/titulo-da-pagina.html>

² <https://static.googleusercontent.com/media/www.google.com/pt-BR//intl/pt-PT/webmasters/docs/guia-otimizacao-para-motores-de-busca-pt-pt.pdf>

- <http://dominio.com/categoria/titulo-da-pagina.html>
- Site deve ser fácil de navegar:
 - É fácil alcançar as opções para navegação?
 - São necessários poucos cliques para se alcançar a informação desejada?
 - Expressões relevantes são interligadas a outras páginas que podem facilitar a compreensão ou acesso a mais informações sobre as mesmas?
 - Há um *sitemap* (em HTML e/ou XML) para facilitar a navegação pelas mesmas?
- Ofereça conteúdo de qualidade:
 - Texto fácil de ler, sem erros ortográficos, gramaticais ou de coerência/coesão;
 - Uso de imagens e/ou vídeos que complementem a informação passada;
 - Organizado em torno do tema principal e respondendo as dúvidas do usuário;
 - Conteúdo deve ser sempre único e original!
 - Sua página deve ser confiável: você colocaria as informações de seu cartão de crédito em um website como aquele?
- Adote bons textos-âncora (texto usado em hyperlinks internos ou externos para vincular o endereço para a página-alvo):
 - Deve incluir a palavra-chave ou variações da mesma;
 - Não exagerar demais no uso das mesmas palavras-chave sempre (super-otimização).
- Use tags de cabeçalho (h1 – h3) adequadamente:
 - A tag h1 geralmente apresenta o título da página ou algo similar;
 - Tags h2 são usadas para iniciar cada seção da página.
- Imagens devem conter atributos alt e title descrevendo-as (se possível, incluindo a palavra-chave ou variação);
- Em hyperlinks para outros websites, use o atributo rel="nofollow" quando não estiver recomendando positivamente ou de forma gratuita. Exemplo:
 - Propagandas e anúncios publicitários;
 - Websites para os quais esteja realizando crítica negativa.
- Adicione frequentemente novos conteúdos (páginas, vídeos etc.) em seu website.
- Inclua em seu website hyperlinks para suas contas nas redes sociais (Facebook, Twitter, Instagram, Pinterest, LinkedIn, Tumblr etc.) e publique as atualizações do mesmo nessas contas;
- Estabeleça relações positivas com outros websites em seu nicho.

Outras dicas

- Monitore o tempo de carregamento de suas páginas;

- Use o planejador de palavras-chave do Google Adwords para identificar outras palavras relacionadas ao seu nicho que lhe sejam úteis;
- Inclua e monitore seu website em:
 - Google Webmaster Tools;
 - Google Analytics;
 - Bing Webmaster Tools.
- Analise o perfil de backlinks para o seu website/página, buscando diversidade e qualidade;
 - Links bons: links em páginas editoriais, em websites relevantes para o seu nicho;
 - Links ruins: links criados de forma automatizada, fora de um contexto (por exemplo, na sidebar ou rodapé) ou em websites em nichos muito diferentes do seu.

O que evitar

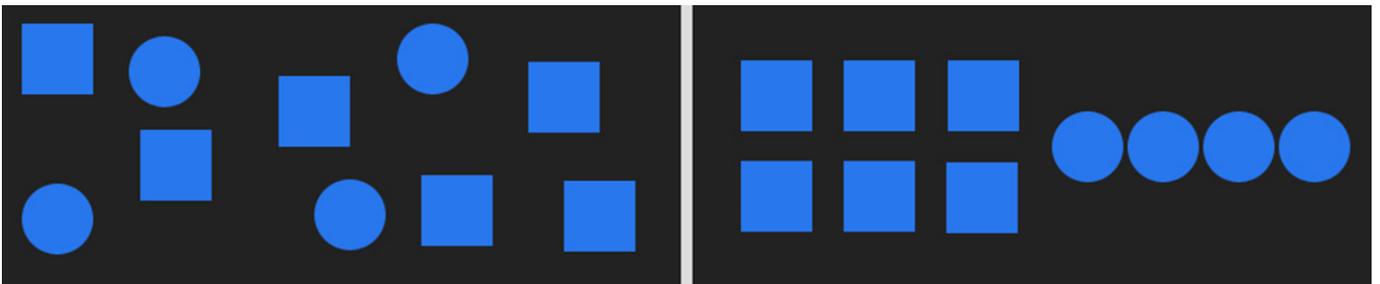
- *Link spamming* (criar links de forma automatizada ou não autorizada em blogs gratuitos, fóruns de discussão, redes sociais etc.);
- Técnicas de *cloaking* (“camuflagem do conteúdo”, isto é, exibir um conteúdo aos motores de busca e outro aos usuários humanos).
- Criar *hyperlinks* para websites “não muito populares” (adultos, cassinos, jogos de azar, com apologia a violência, uso de drogas ou racismo etc.);
- Uso de diversas outras técnicas conhecidas como “*black hat*”;
- Focar demais em motores de busca e esquecer que o importante é conquistar o usuário!

Princípios de Web Design

Princípios de Design³

Alinhamento

- Alinhar é organizar a informação visualmente de forma a apresentar uma aparência menos caótica e mais agradável.



- Campos em formulários;
- Conteúdos em páginas.

Cadastro de Pesquisa

Dados do pesquisador

Nome:

CPF:

Matrícula:

Dados do bolsista

Nome:

CPF:

Matrícula:

Dados da pesquisa

Nome:

Cadastro de Pesquisa

Dados do pesquisador

Nome:

CPF:

Matrícula:

Dados do bolsista

Nome:

CPF:

Matrícula:

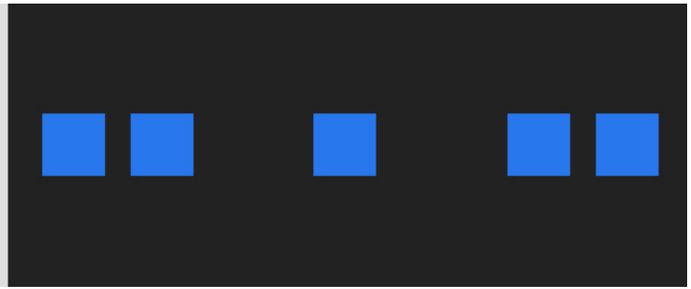
Dados da pesquisa

Nome:

Balanço

- Balancear é distribuir os elementos visualmente de forma que as distâncias entre os mesmos torne-se mais uniforme;
 - Simétrico;
 - Assimétrico;
 - Radial.

³ Livro: Design para quem não é Designer
Artigo online: <http://criativosfera.com/aula-05-principios-do-design/>



- Distribuição mais uniforme das linhas do formulário.

Cadastro de Pesquisa

Dados do pesquisador

Nome:

CPF:

Matrícula:

Dados do bolsista

Nome:

CPF:

Matrícula:

Dados da pesquisa

Nome:

Cadastro de Pesquisa

Dados do pesquisador

Nome:

CPF:

Matrícula:

Dados do bolsista

Nome:

CPF:

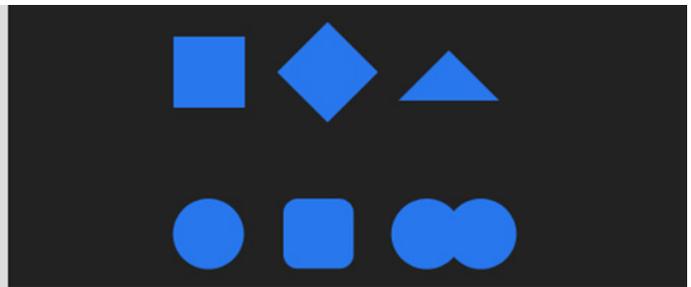
Matrícula:

Dados da pesquisa

Nome:

Proximidade

- A proximidade ou distância entre elementos indica visualmente algum tipo de relação entre os mesmos.



- Distâncias entre objetos que possuem alguma relação e aqueles não relacionados.

Cadastro de Pesquisa

Dados do pesquisador

Nome:

CPF:

Matrícula:

Dados do bolsista

Nome:

CPF:

Matrícula:

Dados da pesquisa

Nome:

Cadastro de Pesquisa

Dados do pesquisador

Nome:

CPF:

Matrícula:

Dados do bolsista

Nome:

CPF:

Matrícula:

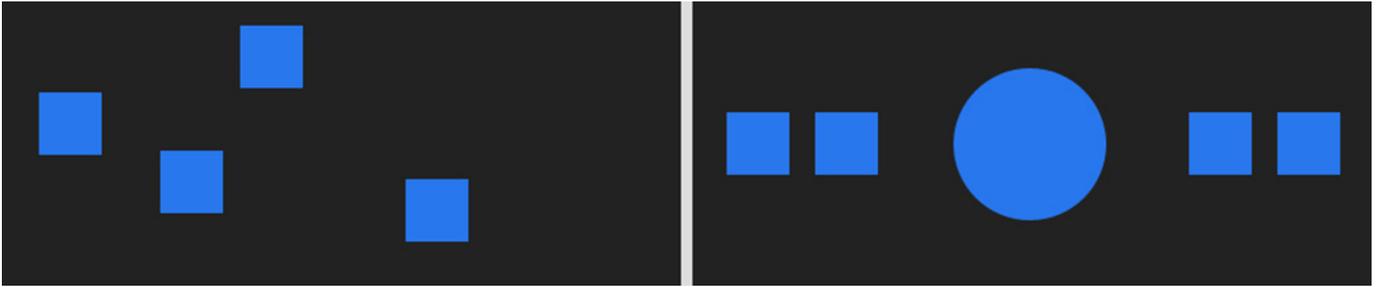
Dados da pesquisa

Nome:

Contraste

- Contrastar é criar algum tipo de diferença visual entre dois ou mais elementos de um grupo;

- Pode-se contrastar alterando o tamanho, cor, tipo da fonte, estilo (negrito, itálico, sublinhado) etc.



- Ênfase a títulos de seção, botões que submetem dados ao servidor e informações importantes para o usuário.

Cadastro de Pesquisa

Dados do pesquisador

Nome:

CPF:

Matrícula:

Dados do bolsista

Nome:

CPF:

Matrícula:

Dados da pesquisa

Nome:

Cadastro de Pesquisa

Dados do pesquisador

Nome:

CPF:

Matrícula:

Dados do bolsista

Nome:

CPF:

Matrícula:

Dados da pesquisa

Nome:

Repetição / Consistência

- Repetir um padrão de layout em vários documentos (ou páginas web) auxilia na criação de uma identidade visual bem como torna mais fácil sua leitura, navegação e manuseio.



- Aplicar mesmo estilo a elementos com funções similares: formulários, labels, botões etc.

The image shows two web forms side-by-side. The left form is titled 'Cadastro de Pesquisa' and is divided into three sections: 'Dados do pesquisador', 'Dados do bolsista', and 'Dados da pesquisa'. Each section has input fields for 'Nome', 'CPF', and 'Matrícula'. The 'Dados da pesquisa' section has a single 'Nome' field. A 'Cadastrar' button is at the bottom. The right form is titled 'Login' and has input fields for 'Usuário' and 'Senha', with a 'Login' button below.

Princípios de Web Design⁴

- Fatores Humano-Computacionais
 - Usabilidade
 - Ergonomia
 - Acessibilidade

Usabilidade

- É a qualidade de um sistema de ser facilmente usado pelo usuário final;
- Indica quão facilmente o usuário conseguirá executar suas tarefas naquele sistema;
- Mínimo de esforço físico e mental possível.

Ergonomia

- É a capacidade de um sistema adaptar-se ao homem para a execução das tarefas – e não o contrário.

Acessibilidade

- É a capacidade de um sistema ser utilizado por qualquer tipo de pessoa, independente de suas limitações;
 - Botões para aumentar a fonte do texto;
 - Layout responsivo;
 - Uso de tags semânticas.

⁴ Livro: Não me Faça Pensar

Artigo online: <https://www.oficinadanet.com.br/post/11073-resenha-do-livro-nao-me-faca-pensar>

Tipos de Websites

Minisites

- Diz-se de todos os tipos de *websites* que apresentam uma pequena quantidade de páginas no total (geralmente, menos de dez páginas);
- Por apresentar poucas páginas e não requerer funcionalidades mais complexas, são geralmente páginas estáticas.

Hotsites

- Minisites utilizados no lançamento de novos produtos, serviços ou promoções;
- Geralmente, apresentam forte apelo visual.

Sites para entretenimento

- Websites criados exclusivamente para fins de lazer, como sites de jogos *web*, vídeos etc.

Sites educacionais

- Oferecem recursos, ferramentas ou plataformas para a educação, seja ela presencial ou a distância.

Blogs, vlogs e flogs

- Websites criados com o intuito de criar registros periódicos (diários ou não) de informações, eventos, pensamentos etc.
- **Blog** - publicações predominantemente no formato texto (podendo incluir vídeos, fotos, etc.);
- **Vlog** - publicações de vídeos;
- **Flog** - publicações de fotografias.

Catálogos, lojas virtuais, sites afiliados e marketplace

- Websites cujo foco é o comércio eletrônico de produtos;
- **Catálogo** – website cujo foco é exclusivamente a exposição dos produtos e serviços da empresa;
- **Loja virtual** – website de uma empresa, focado na venda exclusivamente de seus produtos;
- **Marketplace** – website criado com o objetivo de vender produtos de diversas empresas.

Sites afiliados

- Websites criados com o intuito de vender produtos de terceiros;
- Geralmente capturam a atenção de possíveis consumidores por meio de informações relevantes sobre certos nichos, bem como apresentando revisões de produtos e serviços.

Sites portfólios

- Possuem o objetivo de expor trabalhos (imagens, animações 3D, vídeos, músicas, telas de software etc.) já desenvolvidos, com o intuito de alcançar novos clientes;
- Geralmente criado para pessoas físicas, profissionais autônomos ou pequenas empresas.

Sites institucionais

- Website criado com o intuito de apresentar uma organização e sua cultura organizacional;
- Algumas páginas presentes: apresentação, missão/visão, produtos e serviços, contato;
- Podem ser bem simples (minisites) ou complexos, incorporando blogs e diversos tipos de páginas com finalidades específicas.

Sites comunitários

- Possuem como foco o uso coletivo e compartilhamento de informações. Alguns exemplos são as redes sociais e fóruns de discussão;
- Atualmente, há tanto redes sociais para “uso genérico”, como Facebook e Twitter, como aquelas para uso específico, como Instagram (compartilhamento de imagens a partir de celular) e LinkedIn (para networking profissional).

Sites midiáticos

- Servem para compartilhamento de notícias, reportagens, vídeos e imagens sobre eventos ou fatos ocorridos;
- Muito empregado pelos meios de comunicação em massa.

Sites aplicativos

- Tais *websites* funcionam como aplicativos na nuvem, acessíveis a todos ou mediante login;
- Google Drive (com seu Google Docs, Google Spreadsheet, Google Forms etc.) é um exemplo de site aplicativo (ou “suíte de aplicativos”).

Sites bancos de dados

- Websites que servem para consulta e visualização de informações específicas. Exemplo: para verificar dados no SERASA ou para buscar negócios relacionados a uma área;
- Sites de busca e sites de classificados podem ser considerados subtipos específicos de sites de bancos de dados.

Portais

- Geralmente integram diversos tipos de websites, páginas e conteúdos, com o intuito de atingir um público maior;
- Exemplos de portais são os websites do G1 e UOL.

Parte 4 - JavaScript

Introdução a JavaScript

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.
 - Interpretada – não é compilada, isto é, o código escrito pelo desenvolvedor é lido e executado pelo interpretador (neste caso, o navegador);
 - *Client-side* – é executada no computador do cliente (solicitante), não no servidor⁵;
 - 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;
- 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!
- 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;
- 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;
- Hoje, JavaScript é um padrão em programação client-side, tendo seu uso aliado a outras tecnologias:
 - jQuery
 - JSON
 - Ajax

⁵ Hoje, JS já começa a ser usada no ambiente servidor por meio de bibliotecas/*frameworks* como node.js

Compatibilidade nos *browsers*

- Por ser uma linguagem que sofre poucas alterações (atualmente na versão 1.8.5⁶, de março de 2011), maioria dos novos browsers suporta (quase) todas suas características.

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.
- 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

[Ver exemplo 1.1]

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

Forma #2 – dentro de uma <script> na página

[Ver exemplo 1.2]

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

Forma #3 – em um arquivo .js

[Ver exemplo 1.3]

6 Fonte: <https://pt.wikipedia.org/wiki/JavaScript>

exemplo-1.3.html <code><script src="exemplo-1.3.js"></script></code>	exemplo-1.3.js <code>alert('Olá');</code>
--	---

Depurando scripts

[Ver exemplo 1.4]

- 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>

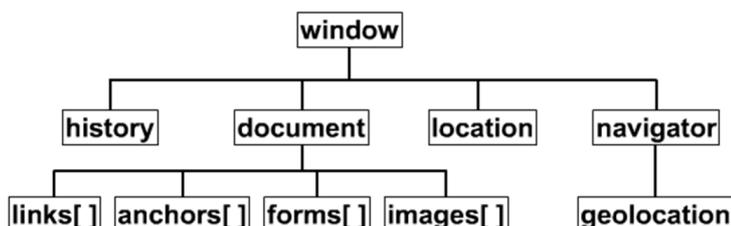
Document Object Model e Eventos

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 window⁷

[Ver exemplo 2.1]

Propriedade	Descrição
name	Nome da janela
parent	Janela “mãe”
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

7 Fonte: https://www.w3schools.com/jsref/obj_window.asp

moveTo(X, Y)	Move a janela
--------------	---------------

Objeto document⁸

[Ver exemplo 2.2]

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 ¹⁰	Dados do cookie do documento atual

Método	Descrição
write(...)	Escreve os valores especificados no documento
getElementById(...) ⁹	Retorna o elemento cujo <i>id</i> foi especificado

Objeto location¹¹

[Ver exemplo 2.3]

Propriedade	Descrição
href	URL da página
protocol	Protocolo (http, https, ftp 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

Objeto history¹²

[Ver exemplo 2.4]

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

⁸ Fonte: https://www.w3schools.com/jsref/dom_obj_document.asp

⁹ Também é possível pegar um HTML Element apenas usando seu id, sem o uso desta função

¹⁰ Fonte: https://www.w3schools.com/js/js_cookies.asp

¹¹ Fonte: https://www.w3schools.com/jsref/obj_location.asp

¹² Fonte: https://www.w3schools.com/jsref/obj_history.asp

go(X)	Volta (valores negativos) ou avança (valores positivos) para uma URL da lista
-------	---

Objeto navigator¹³

[Ver exemplo 2.5]

Propriedade	Descrição
appName	Codigono do navegador
appName	Nome do navegador
appVersion	Versão do navegador
cookieEnabled	Verifica se os cookies estão habilitados
geolocation ¹⁴	Objeto <i>geolocation</i> (coordenadas)
onLine	Verifica se o navegador está online

Método	Descrição
javaEnabled()	Verifica se Java está habilitado

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 eventos¹⁵

[Ver exemplo 2.6]

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

Principais eventos (mouse)

[Ver exemplo 2.7]

13 Fonte: https://www.w3schools.com/jsref/obj_navigator.asp

14 Saiba mais em: https://www.w3schools.com/html/html5_geolocation.asp

15 Fonte: https://www.w3schools.com/jsref/dom_obj_event.asp

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 dos 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)

[Ver exemplo 2.8]

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 event¹⁶

[Ver exemplo 2.8]

Propriedade	Descrição
type	Tipo do evento ocorrido
target	Objeto relacionado ao evento
which / keyCode ¹⁷	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

¹⁶ Fonte: https://www.w3schools.com/jsref/dom_obj_event.asp

¹⁷ Alguns navegadores implementam o atributo which, outros, keyCode

Tipos de Dados, Variáveis e 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.

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.

- 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

[Ver exemplo 3.1]

Função	Descrição	Exemplo
eval ¹⁸	Avalia a string passada	eval("2 + 2") eval("alert('Teste')")
parseInt	Converte string em inteiro	parseInt("12.3")
parseFloat	Converte string em número com decimais	parseFloat("12.3")
Number	Converte objeto em número	Number(true)
String	Converte objeto em string	String(true)
N.toString()	Converte o número em string	n.toString()

18 **Atenção!** https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/eval#Never_use_eval!

		(13.105).toString()
N.toFixed(X)	Converte o número em string com X casas decimais	n.toFixed(2) (13.105).toFixed(2)

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

[Ver exemplo 3.2]

- 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

[Ver exemplo 3.3]

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

Atribuições

Operador	Significado	Exemplo
=	Recebe (armazena um valor)	x = 3

<code>+=</code>	Recebe o valor dele adicionado com	<code>x += 3</code>
<code>-=</code>	Recebe o valor dele subtraído dele	<code>x -= 3</code>
<code>*=</code>	Recebe o valor dele multiplicado por	<code>x *= 3</code>
<code>/=</code>	Recebe o valor dele dividido por	<code>x /= 3</code>
<code>%=</code>	Recebe o resto da divisão dele por	<code>x %= 3</code>

Operadores relacionais

[Ver exemplo 3.3]

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

Operadores lógicos

Operador	Significado	Exemplo
<code>&&</code>	E	<code>x && y</code>
<code> </code>	Ou	<code>x y</code>
<code>!</code>	Não	<code>!x</code>

Operadores para string

Operador	Significado	Exemplo
<code>+</code>	Concatenação	<code>x + y</code>

Operadores especiais

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

Funções e comandos de controle e 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

[Ver exemplo 4.1]

//Muda cor de fundo da página

```
function mudaCor(cor) {  
    document.bgColor = cor;  
}
```

//Calcula e retorna a média de dois números

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

//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

[Ver exemplo 4.2]

//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

[Ver exemplo 4.2]

//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

[Ver exemplo 4.3]

```
//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

[Ver exemplo 4.3]

```
//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

[Ver exemplo 4.3]

[//Imprime as propriedades do objeto usuario](#)

```
var usuario = { nome: "Christiano",  
               matricula: "001",  
               curso: "Informática"};  
  
for (var prop in usuario) {  
    document.write(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

Objetos

- 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!
```

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";
```

Sintaxe #2

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

Exemplo

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

Definindo métodos

Sintaxe #1

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

Exemplo

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

Sintaxe #2

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

Exemplo

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

Arrays

- 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!
```

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 e Métodos de um Array

Propriedade	Descrição
length	Retorna o tamanho do array

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

Date, Math, String e Object

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);
```

Métodos de um objeto 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

- Apresenta um conjunto de propriedades e métodos úteis em operações matemáticas.

Propriedades e Métodos de 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(num2, ...)	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

Propriedades e Métodos de um objeto 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(antiga, nova)	Substitui a ocorrência da substring antiga pela substring nova
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

Object

Propriedades e Métodos de Object

Propriedade	Descrição
Object.prototype*	Representa o 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]

Parte 5 - Bootstrap

Introdução ao Bootstrap

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;  
  
}  
  
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 (versão CDN):

```
<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>
```

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>
```

- **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

[Ver exemplo 7.1]

```
<!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

[Ver exemplo 7.2]

- 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

Ícones

[Ver exemplo 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 (versão CDN):

```
<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>
```

Saiba mais em: https://www.w3schools.com/icons/fontawesome_icons_intro.asp

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 classes.

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

Exemplos de uso:

- Exemplo 1 [Ver exemplo 8.2]:

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

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

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

- Exemplo 2 [Ver exemplo 8.3]:

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

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

Renderizando imagens

[Ver exemplo 8.4]

- 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}`

Exemplos:

- Exemplo 1:

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

Esta div ocupará 10 colunas, sempre

```
</div>
```

- Exemplo 2:

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

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

</div>

- Exemplo 3:

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

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

</div>

- Exemplo 4:

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

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

</div>

- Exemplo 5:

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

Esta div ocupará 10 colunas, mas se o dispositivo for "sm", ocupará 2

</div>

Exemplos de página com três colunas

[Ver exemplo 8.5]

- Exemplo 1 [Ver exemplo 8.5]:

```
<div class="row">
```

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

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

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

</div>

- Exemplo 2 (layout mais responsivo) [Ver exemplo 8.6]:

```
<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! [Ver exemplo 8.7]

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

Alguns Componentes no Bootstrap

Barra de navegação

- Muitos *web* 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>
```

- 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 / transparente.

Exemplo [Ver exemplo 9.1]:

```
<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**).

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

[Ver exemplo 9.2]

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

- 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>.

[Ver exemplo 9.3]

```
<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>
```

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

[Ver exemplo 9.4]

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

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

[Ver exemplo 9.5]

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

[Ver exemplo 9.6]

```
<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.

Exemplo de card simples

[Ver exemplo 9.7]

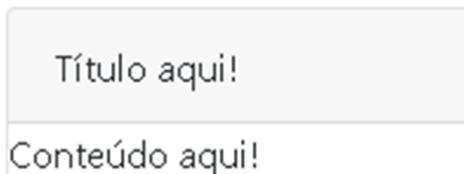
```
<div class="card">
  Conteúdo aqui!
</div>
```



Exemplo de card com cabeçalho

[Ver exemplo 9.8]

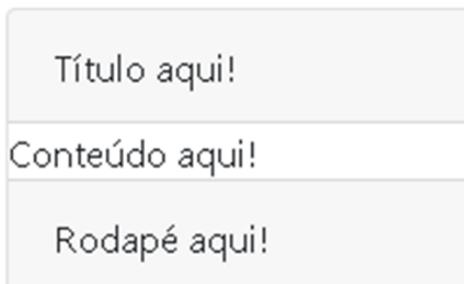
```
<div class="card">
  <div class="card-header">
    Título aqui!
  </div>
  Conteúdo aqui!
</div>
```



Exemplo de card com cabeçalho e rodapé

[Ver exemplo 9.9]

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

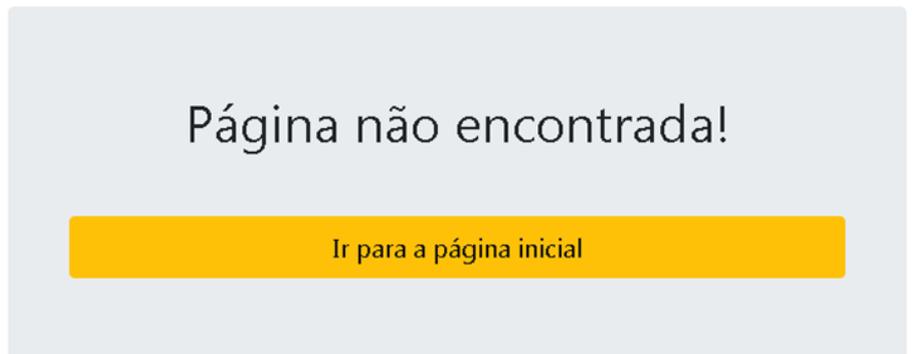


Vejamos como fica em nossa página... [Ver exemplo 9.10]

Jumbotron

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

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



- O jumbotron pode ser usado fora de um container... [Ver exemplo 9.11]
- ... ou dentro de um. [Ver exemplo 9.12]

Vamos usar um para exibir um anúncio promocional em nossa página. [Ver exemplo 9.13]

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:

```
<span class="badge">Valor</span>
```

- Exemplo de badge para carrinho de compras:

[Ver exemplo 9.14]

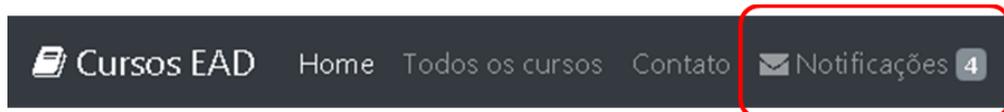
```
<a href="carrinho.html">
  <span class="fa fa-shopping-cart"></span>
  <span class="badge badge-primary">4</span>
</a>
```



- Adicionando um badge à barra de navegação:

[Ver exemplo 9.15]

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

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

[Ver exemplo 9.16]

```
<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:

[Ver exemplo 9.17]

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



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.).

Exemplo [Ver exemplo 9.18]:

```
<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 Bar

- A breadcrumb navigation bar (barra de 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.

Exemplo [Ver exemplo 9.19]:

```
<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”.

Exemplo [Ver exemplo 9.20]:

```
<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>
```

«	1	2	3	»
---	---	---	---	---

Formatando tabelas e 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.

[Ver exemplo 10.1]

```
<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>
```

- Agora, aplique cada uma das seguintes classes à tag table (uma por uma e de forma acumulada) e descreva o resultado [Ver exemplo 10.2]:
 - table
 - table-bordered
 - table-striped
 - table-hover
 - table-responsive
- Agora, experimente as opções de cores (classes) thead-light e thead-dark para a tag thead [Ver exemplo 10.3].

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?

[Ver exemplo 10.4]

```

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

```

- 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”.
- Envolve cada input/textarea com um div contendo a classe “col-sm-10 input-group” [Ver exemplo 10.5].
- Inclua antes de cada input/textarea (dentro do div) um dos seguintes spans:
 -
 - @
 -
- Envolve o botão com uma div com as classes “col-sm-12 text-center” e inclua o seguinte span antes do texto “Enviar” [Ver exemplo 10.6]:
 -

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

- 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.
- **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>
```

- **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>
```

Exemplo de corpo:

```
<div class="modal-body">
  Conteúdo da janela.
</div>
```

Exemplo de rodapé:

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

```
</div>
```

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

[Ver exemplo 11.1]

```
<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.

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

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

```
    ...
```

```
</div>
```

- **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>
```

- **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>
```

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

[Ver exemplo 11.2]

```
<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>
```

- **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:

[Ver exemplo 11.3]

```
<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

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">
```

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

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

```
...
```

```
</div>
```

- **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>
```

- **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/baguetteBox.min.js"></script>
```

```
<script>
```

```
  baguetteBox.run('.gallery');
```

```
</script>
```