

Módulo de Desenvolvimento

HTML - CSS - JAVASCRIPT - BOOTSTRAP

Professores:
Prof. Humberto Delgado
Prof. Ricardo Alexandre Bontempo

TÓPICOS

1

HTML

5

Javascript

2

Formulário

3

CSS

4

BootStrap

CONFIGURANDO O AMBIENTE PARA AS PÁGINAS

Quando vamos construir páginas para uma aplicação web, temos algumas etapas que precisamos nos lembrar:

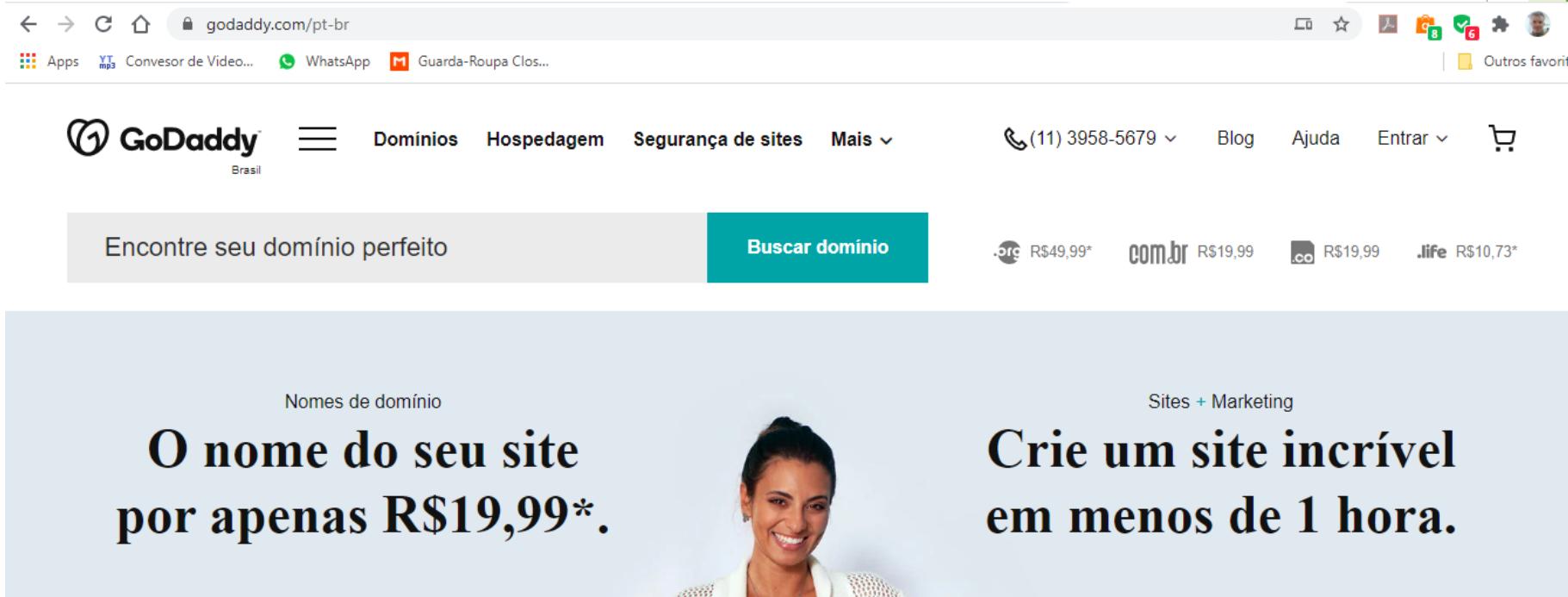
1º Criar um endereço para a página

Deveremos registrar a página e para isso podemos utilizar o site: Registro.br



2º Após obter o endereço teremos que hospedar a nossa página

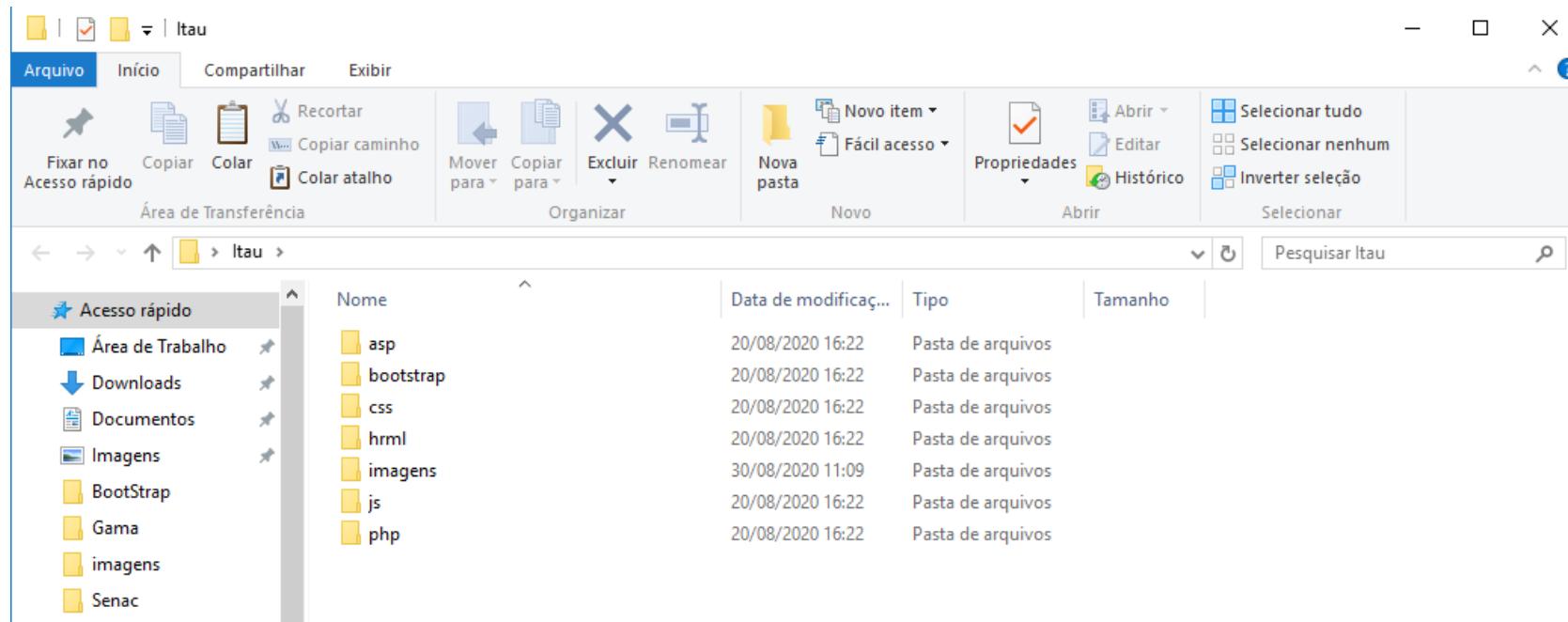
Existem hoje várias plataformas, como por exemplo: locaweb.com.br, uol.com.br, godaddy.com/pt-br entre outras



The screenshot shows the GoDaddy website homepage. At the top, there's a navigation bar with links for 'Domínios', 'Hospedagem', 'Segurança de sites', 'Mais', 'Blog', 'Ajuda', 'Entrar', and a shopping cart icon. Below the navigation, there's a search bar with the placeholder 'Encontre seu domínio perfeito' and a 'Buscar domínio' button. To the right of the search bar, there are price offers for different domain extensions: '.org' for R\$49,99*, '.com.br' for R\$19,99, '.co' for R\$19,99, and '.life' for R\$10,73*. On the left side, there's a large banner with the text 'Nomes de domínio' and 'O nome do seu site por apenas R\$19,99*.' On the right side, there's another banner with the text 'Sites + Marketing' and 'Crie um site incrível em menos de 1 hora.' There's also a portrait of a smiling woman.

3º Neste momento é muito bom criar as pastas aonde iremos armazenar todas as páginas, imagens, estilos entre outros procedimentos.

Exemplo:



OBS: Este procedimento permite depois fazer o Upload para o site de hospedagem ou servidor

Dicas úteis

Nome da Página: **Index.html**

Como regra, o arquivo chamado index é a página inicial padrão. Isso significa que esta página será a primeira a ser exibida quando você acessar o seu domínio diretamente (www.exemplo.com). Além disso, index.php será sempre exibido ante de index.html.

Tipos de Imagens: **Imagens**

Por padrão hoje temos três tipos de imagens que são utilizadas no desenvolvimento de páginas na web:

JPEG, GIF e PNG

Tipos de Imagens: Imagens

Formato JPEG (JPG)

O formato JPEG, cuja sigla significa Joint Photographic Experts Group, teve sua primeira especificação disponibilizada em 1983 por um grupo que leva o mesmo nome. É um dos padrões mais populares da internet por aliar duas características importantes: oferece níveis razoáveis de qualidade de imagem e gera arquivos de tamanho pequeno quando comparado a outros formatos, facilitando o seu armazenamento e a sua distribuição.



Formato GIF

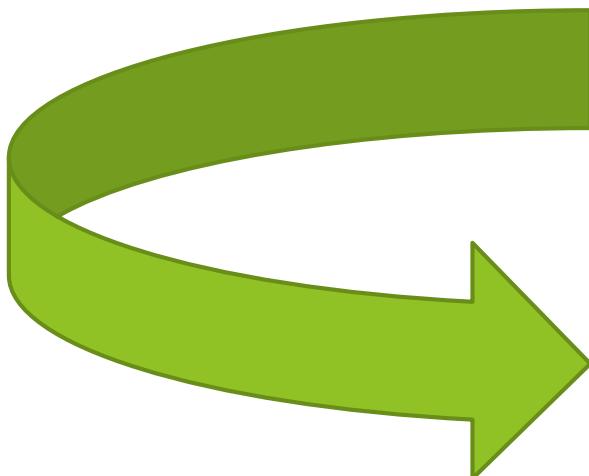
Sigla para Graphics Interchange Format, o GIF é outro formato bastante popular na internet. Foi criado pela CompuServe em 1987 e, assim como o JPEG, gera arquivos de tamanho reduzido, no entanto, seu uso não é muito comum em fotografias, já que é capaz de trabalhar com apenas 256 cores (8 bits). Por este motivo, sua utilização é muito frequente com ícones, ilustrações ou qualquer tipo de imagem que não necessite de muitas cores.



Tipos de Imagens: Imagens

Formato PNG

O formato PNG, sigla para Portable Network Graphics, é um dos padrões mais recentes, com a sua primeira especificação tendo surgido em 1996. Seu desenvolvimento foi motivado, em parte, pela restrição de patente existente no formato GIF, conforme explica o tópico anterior. O PNG reúne, portanto, as características que tornaram o GIF tão bem aceito: animação, fundo transparente e compressão sem perda de qualidade, mesmo com salvamentos constantes do arquivo. Porém, conta com um grande diferencial: suporta milhões de cores, não apenas 256, sendo, com isso, uma ótima opção para fotos.



INTRODUÇÃO AO HTML

HTML - HYPERTEXT MARKUP LANGUAGE

É uma linguagem de marcação utilizada para formatações de textos, inserir imagens e ligações de hipertexto.

Não é possível programar em linguagem HTML pois ela é simplesmente uma linguagem de marcação.

Os navegadores (browsers) são os responsáveis por identificar as marcações em HTML e apresentar os documentos conforme o que foi especificado por essas marcações.

HTML – ESTRUTURAS BÁSICAS

Todo documento HTML apresenta elementos entre parênteses angulares (< e >) - esses elementos são as etiquetas (tags) de HTML, elas definem onde começa e termina um determinado conteúdo na página. A **maioria** das tags tem abertura e fechamento: <tag>...</tag>

Alguns elementos são chamados "vazios", pois não marcam uma região de texto, apenas inserem algo no documento: <tag>

Todos os elementos podem ter **atributos**: <tag atributo1="valor1" atributo2="valor2">...</tag>

Os atributos podem ser obrigatórios ou opcionais. Use sempre **minúsculas** para escrever as tags

HTML – ESTRUTURAS BÁSICAS

As tags que compõe a estrutura básica de um código HTML são:

<!DOCTYPE html> Deve ser a primeira linha de código do documento antes da tag HTML.

O Doctype indica para o navegador e para outros meios qual a especificação de código utilizar. Em versões anteriores, era necessário referenciar o DTD diretamente no código do Doctype. Com o HTML5, a referência por qual DTD utilizar é responsabilidade do Browser. O Doctype não é uma tag do HTML, mas uma instrução para que o browser tenha informações sobre qual versão de código a marcação foi escrita.

<html> Define o início do documento HTML.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Insert title here</title>.
6 </head>
7 <body>
8
9 </body>
10 </html>
```

HTML – ESTRUTURAS BÁSICAS

As tags que compõe a estrutura básica de um código HTML são:

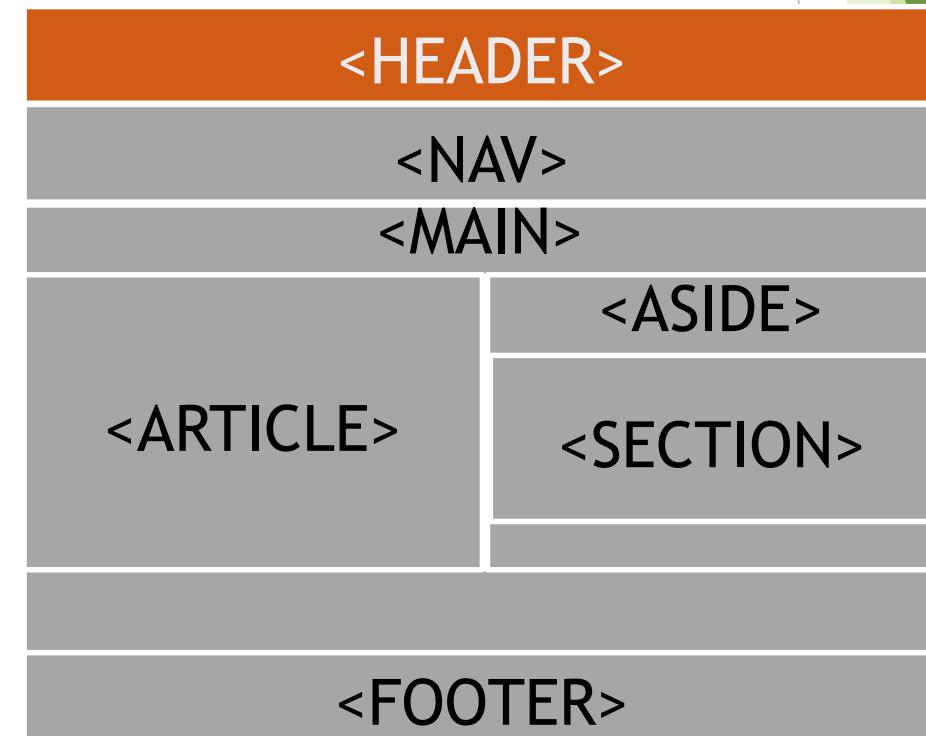
<head> É onde fica toda uma parte muito importante de sua página. No HEAD ficam os metadados. Metadados geralmente definem o título do documento, o conjunto de caracteres, os estilos e outras metainformações.

<meta charset="ISO-8859-1"> Para exibir uma página HTML corretamente, um navegador da web deve conhecer o conjunto de caracteres usado na página. Isso é especificado na tag <meta>.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Insert title here</title>
6 </head>
7 <body>
8
9 </body>
10 </html>
```

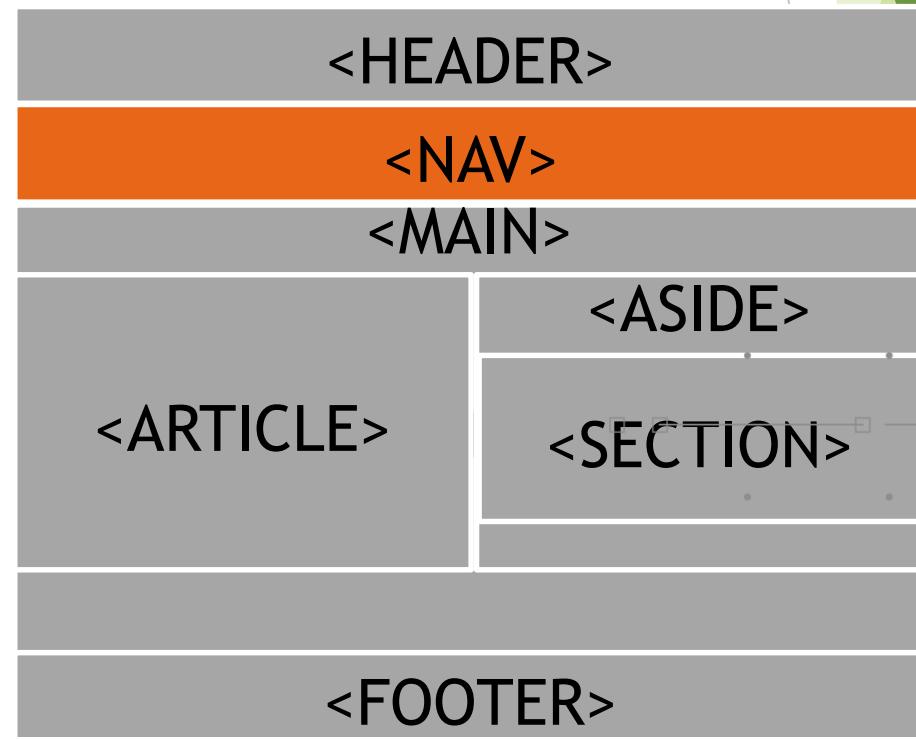
HTML – ESTRUTURAS BÁSICAS - REFORÇANDO

O elemento header representa um grupo de introdução ou elementos de navegação. O elemento header pode ser utilizado para agrupar índices de conteúdos, campos de busca ou até mesmo logos.



HTML – ESTRUTURAS BÁSICAS - REFORÇANDO

O elemento nav representa uma seção da página que contém links para outras partes do website. Nem todos os grupos de links devem ser elementos nav, apenas aqueles grupos que contém links importantes. Isso pode ser aplicado naqueles blocos de links que geralmente são colocados no Rodapé e também para compor o menu principal do site.



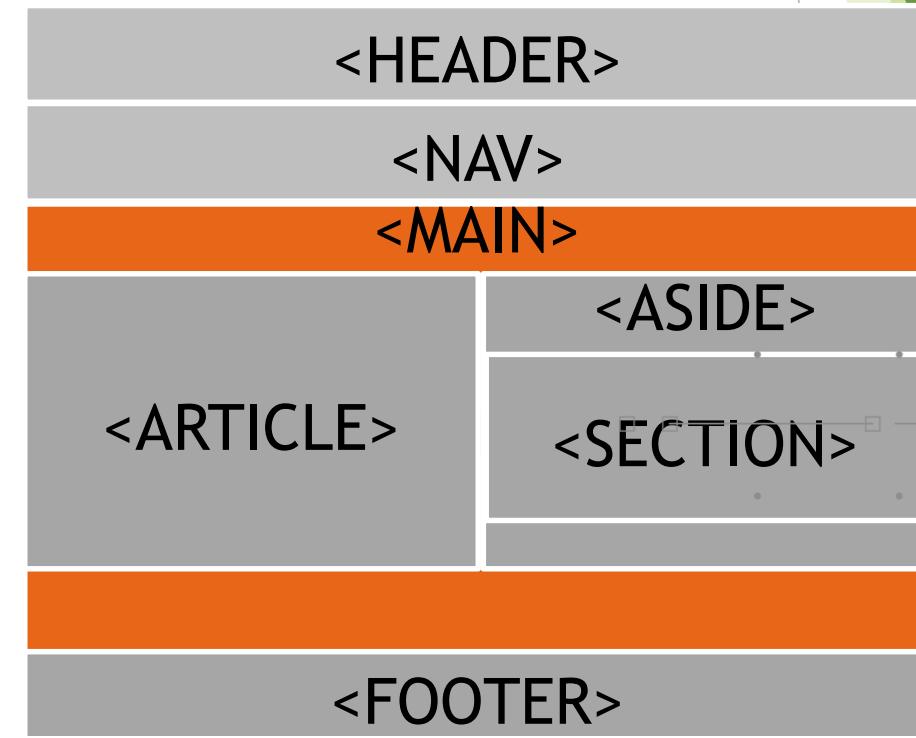
HTML – ESTRUTURAS BÁSICAS - REFORÇANDO

Defini o conteúdo principal da página ou da aplicação. Ele representa o conteúdo mais importante da página, que está diretamente relacionado ao tópico central do documento ou a funcionalidade principal de uma aplicação.

Um detalhe importante: você não pode colocar mais do que UMA tag main no seu documento.

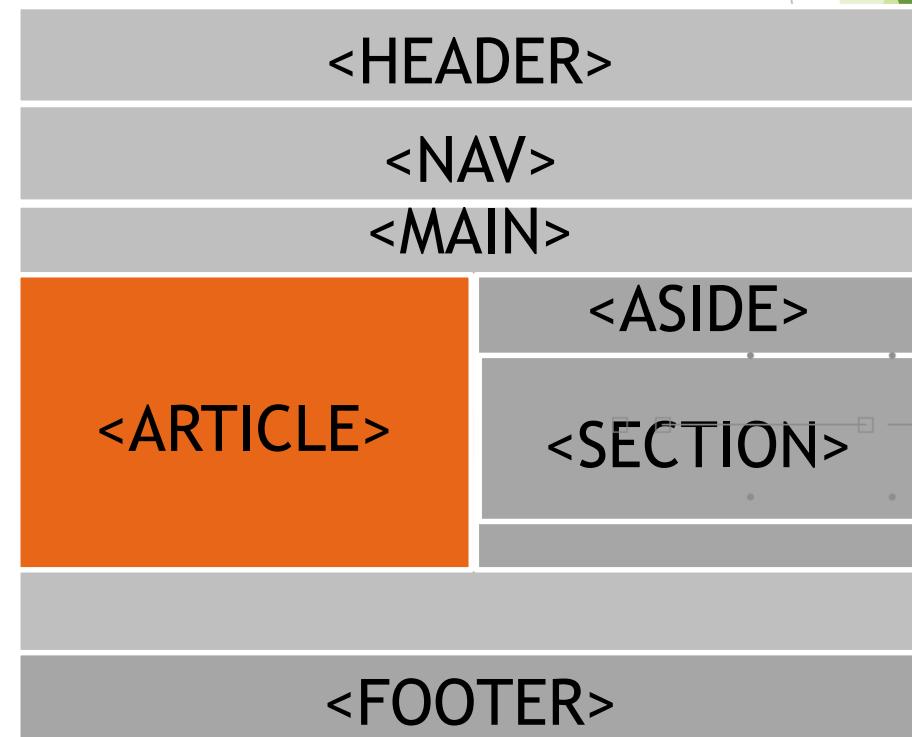
Alguns browsers ainda não suportam esta nova TAG, precisa, renderizar utilizando.

display: block;



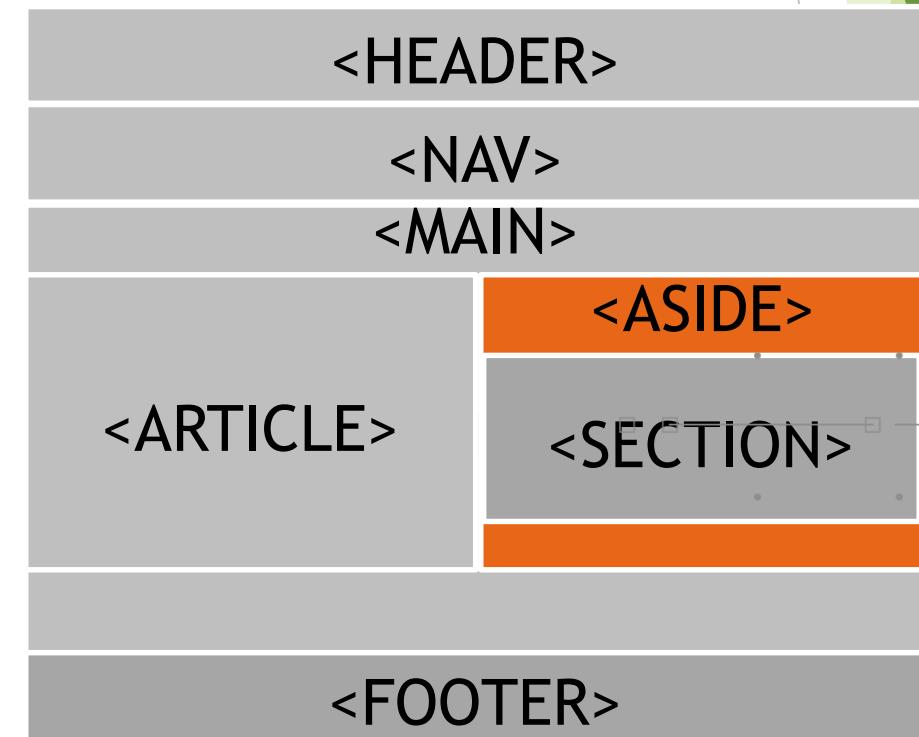
HTML – ESTRUTURAS BÁSICAS - REFORÇANDO

O elemento article representa uma parte da página que poderá ser distribuído e reutilizável em FEEDs por exemplo. Isto pode ser um post, artigo, um bloco de comentários de usuários ou apenas um bloco de texto comum.



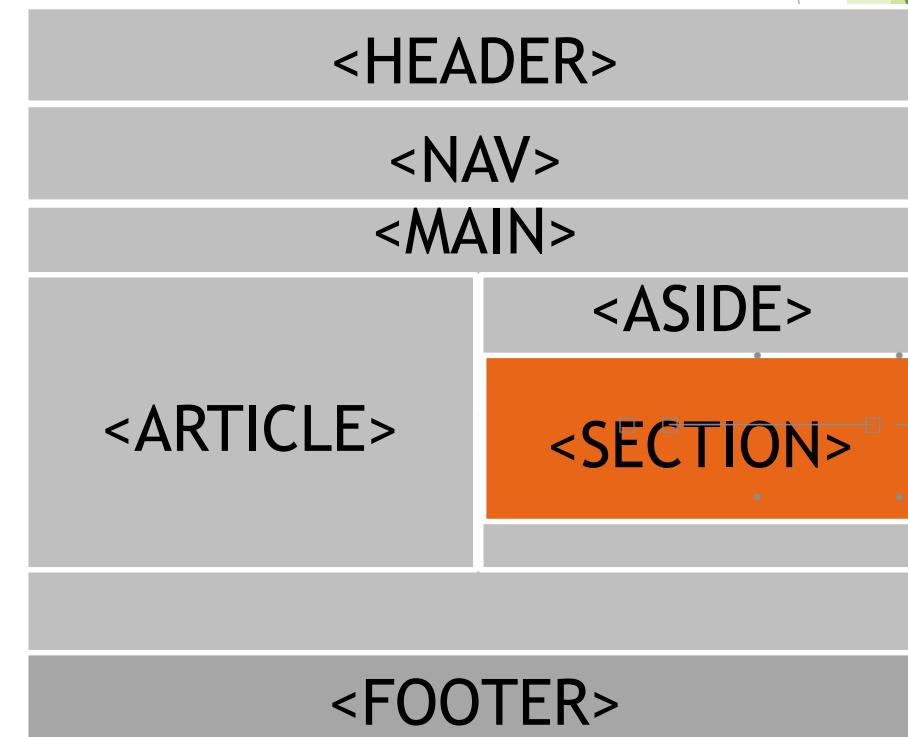
HTML – ESTRUTURAS BÁSICAS - REFORÇANDO

O elemento aside representa um bloco de conteúdo que referencia o conteúdo que envolta do elemento aside. O aside pode ser representado por conteúdos em sidebars em textos impressos, publicidade ou até mesmo para criar um grupo de elementos nav e outras informações separados do conteúdo principal do website.



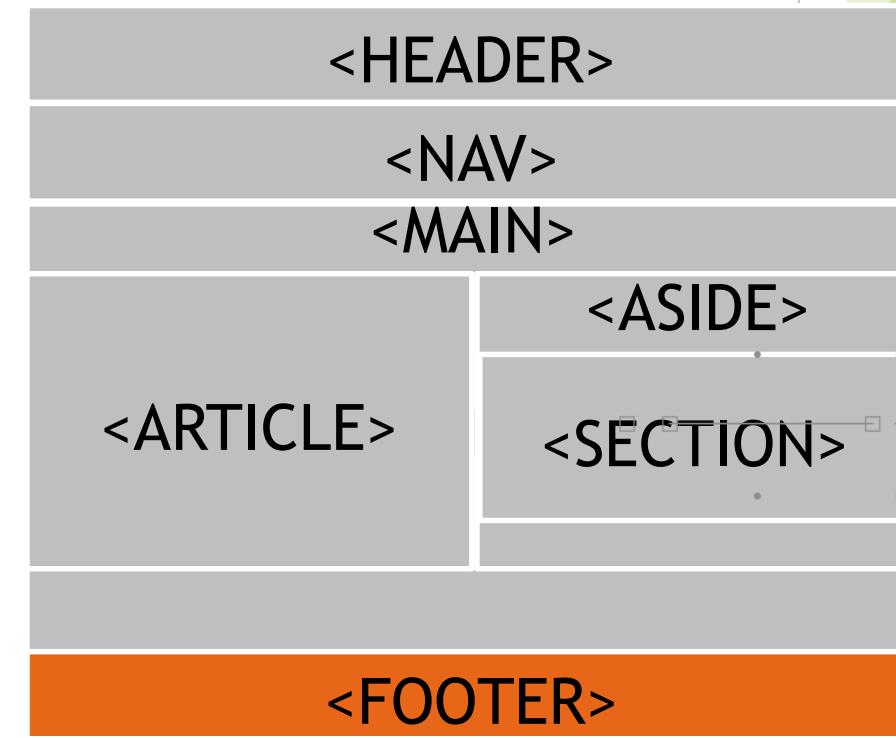
HTML – ESTRUTURAS BÁSICAS - REFORÇANDO

A tag section define uma nova seção genérica no documento. Por exemplo, a home de um website pode ser dividida em diversas seções: introdução ou destaque, novidades, informação de contato e chamadas para conteúdo interno.



HTML – ESTRUTURAS BÁSICAS - REFORÇANDO

O elemento footer representa literalmente o rodapé da página. Seria o último elemento antes de fechar a tag HTML. O elemento footer não precisa aparecer necessariamente no final de uma seção.



HTML – ESTRUTURAS BÁSICAS

As tags que compõe a estrutura básica de um código HTML são:

<title> Define o título da página, o conteúdo que será exibido nas abas dos navegadores.

<body> É aqui que todo conteúdo de sua página será inserido. Entenda como conteúdo: textos, imagens, links, tabelas, formulários, tudo aquilo que será exibido ao usuário.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Insert title here</title>
6 </head>
7 <body>
8
9 </body>
10 </html>
```

HTML – ESTRUTURAS BÁSICAS - REFORÇANDO

Mais tags que compõe a estrutura básica de um código HTML são:

**
** Quebra uma linha

<p> Inicia um parágrafo

<div></div> define um bloco de elementos na pagina HTML

**** em conjunto do estilo é possível formatar um texto. Ela por si só não faz alteração alguma no código.

<address></address> define informações de contato do proprietário/autor da pagina.

HTML – FORMATAÇÃO DE TEXTO

As tags que auxiliam na formatação do texto são:

`<i>texto sendo alterado para a forma itálica</i>`

`texto alterado para negrito`

` ênfase em negrito `

`<u>texto sublinhado</u>`

`²` (texto elevado)

`_{dinâmico}` (texto rebaixado)

`<strike>texto tachado</strike>`

HTML – CABEÇALHOS

As tags para cabeçalho são:

- `<h1></h1>`
 - `<h2></h2>`
 - `<h3></h3>`
 - `<h4></h4>`
 - `<h5></h5>`
 - `<h6></h6>`
- O que muda entre elas é o tamanho do texto e o espaçamento entre as linhas.

HTML – LISTAS

Tags:

**** type="I" **** para criar listas ordenadas (ol = ordered list). Esta lista pode ser alfabética ou numérica.

**** cria listas não ordenadas (ul = unordered list).

Esta será uma lista somente de bullets (marcações).

**** para criar os itens da listas (ol e ul)

HTML – LISTAS

Exemplos:

```
1 <html>
2   <head>
3     <title>Lista ordenadas</title>
4   </head>
5   <body>
6     <ol type = "A">
7       <li>HTML</li>
8       <li>Download</li>
9       <li>WEB</li>
10    </ol>
11  </body>
12 </html>
```

```
1 <html>
2   <head>
3     <title>Lista não ordenada</title>
4   </head>
5   <body>
6     <ul type="square">
7       <li>HTML</li>
8       <li>Download</li>
9       <li>WEB</li>
10    </ul>
11  </body>
12 </html>
```

```
1 <html>
2   <head>
3     <title>Lista não ordenada</title>
4   </head>
5   <body>
6     <ul type="circle">
7       <li>HTML</li>
8       <li>Download</li>
9       <li>WEB</li>
10    </ul>
11  </body>
12 </html>
```

HTML – IMAGENS

**** Imagens são inseridas pela tag . Possui o atributo src que indica o local onde a imagem está armazenada e o atributo alt que auxiliará na usabilidade da página com pessoas com algum tipo de deficiência visual.

```

```

OBS.: se a imagem não for carregada, o navegador exibirá o valor inserido no atributo alt.

HTML – LINKS

O Hipertexto é o responsável pelo grande sucesso da internet, possibilitando a navegação entre arquivos de imagens, texto, áudio, vídeos, etc. São representadas pelas TAGS: `<a>`

`âncora`

Possui atributos do tipo:

href especifica a referência do arquivo a ser localizado

name representa o nome da âncora

target define o destino do vínculo (_blank, _self, _parent, _top, etc...)

_blank = Documento vinculado em uma nova janela

_self = Abre o documento vinculado no mesmo quadro, uma vez que foi clicado

_parent = abre o documento vinculado no quadro pai

_top = abre o documento vinculado em todo o corpo da janela

HTML – LINKS

Exemplos:

Link Absoluto

```
<a href="http://www.itau.com.br">site do Itau</a>
```

Link Relativo

```
<a href="educacao/pagina.html">Página educação</a>
```

Link com Imagem

```
<a href="imagem/fotoAluno.jpg">Foto do aluno</a>
```

HTML – LINKS

Exemplos:

Link com Javascript

```
<a href="javascript:alert('Meu Site!')">clique aqui</a>
```

Link para email

```
<a href="mailto:meuemail@itau.com.br">Fale Conosco</a>
```

Link para email com assunto

```
<a href="mailto:fulano1@itau.com.br?subject=Aula de HTML">Fale Conosco</a>
```

HTML – LINKS

Exemplos:

Link para e-mail com cópia

```
<a href="mailto:fulano1@itau.com.br&cc=eu@itau.com.br">Fale Conosco</a>
```

Link para e-mail com corpo do e-mail

```
<a href="mailto:fulano1@itau.com.br&body=Deixe aqui seu texto">Fale Conosco</a>
```

HTML – LINKS

Links para partes da página

Criar um link no final da página para retornar junto do elemento H1 localizado no topo.

```
<h1 id="topo">Aqui começa a página</h1>
```

...

...

...

...

```
<p><a href="#topo">Topo da página</a></p>
```

HTML – TABELAS

Tags:

`<table></table>` delimita a área de uma tabela.

» `border=[pixel]` – Especifica o tamanho da borda da tabela

» `cellpadding=[pixel]` – Especifica o espaço entre a borda da célula e seu conteúdo.

» `cellspacing=[pixel]` – Especifica o espaço entre as células

» `width=[pixel ou %]` – Especifica a largura da tabela

`<caption></caption>` Legenda da tabela

HTML – TABELAS

Tags:

<thead></thead> determina a “cabeçalho” da tabela

<tbody></tbody> determina o “corpo” da tabela

<tfoot></tfoot> determina o “rodapé” da tabela

– Propriedade idênticas para as tags acima.

» align=[value] – Especifica o alinhamento horizontal

» valign=[value] – Especifica o alinhamento vertical

HTML – FORMATAÇÃO DE TEXTO

Tags:

`<th></th>` determina a coluna do “cabeçalho” e “rodapé” da tabela

» `align=[value]` – Especifica o alinhamento horizontal

» `valign=[value]` – Especifica o alinhamento vertical

» `colspan=[numero]` – Especifica a quantidade de colunas que serão mescladas

» `rowspan=[numero]` – Especifica a quantidade linhas que serão mescladas.

HTML – TABELAS

Tags:

`<th></th>` determina a coluna do “cabeçalho” e “rodapé” da tabela

» `align=[value]` – Especifica o alinhamento horizontal

» `valign=[value]` – Especifica o alinhamento vertical

» `colspan=[numero]` – Especifica a quantidade de colunas que serão mescladas

» `rowspan=[numero]` – Especifica a quantidade linhas que serão mescladas.

HTML – TABELAS

Tags:

`<tr></tr>` determina a linha do “cabeçalho”, “corpo”, “rodapé”

`<td></td>` determina a conteúdo do “corpo” da tabela

– Propriedades idênticas para as tags acima.

» `align=[value]` – Especifica o alinhamento horizontal

» `valign=[value]` – Especifica o alinhamento vertical

» `colspan=[numero]` – Especifica a quantidade de colunas que serão mescladas

» `rowspan=[numero]` – Especifica a quantidade linhas que serão mescladas.

HTML – TABELAS

Exemplo: monte o código para a tabela abaixo:

Legenda da Tabela

Primeira linha, primeira coluna	Primeira linha, segunda coluna	Primeira linha, terceira coluna
Segunda linha, primeira coluna	Segunda linha, segunda coluna	Segunda linha, terceira coluna

HTML – TABELAS

Solução:

```
8<body>
 9<table border="1" cellpadding="5" cellspacing="2" width="80%" align="center">
10 <caption>Legenda da Tabela</caption>
11 <tr>
12 <td>Primeira linha, primeira coluna</td>
13 <td>Primeira linha, segunda coluna</td>
14 <td>Primeira linha, terceira coluna</td>
15 </tr>
16 <tr>
17 <td>Segunda linha, primeira coluna</td>
18 <td>Segunda linha, segunda coluna</td>
19 <td>Segunda linha, terceira coluna</td>
20 </tr>
21 </table>
22 </body>
```

HTML – CORES

Observe as regras de estilo a seguir:

- 1) body {background-color: #FF0000;}
- 2) body {background-color: #F00;}
- 3) body {background-color: rgb(255, 0, 0);}
- 4) body {background-color: rgb(100%, 0%, 0%);}
- 5) body {background-color: red;}

$\#FF0000 = \#F00 = \text{rgb}(255,0,0) = \text{rgb}(100\%,0\%,0\%) = \text{red}$

HTML – CORES

Cor pelo Hexadecimal

Esta é a maneira mais conhecida de definir uma cor.

Convém ressaltar que em uma regra CSS é indiferente usar letras maiúsculas ou minúsculas na sintaxe hexadecimal de cores e também que é válido abreviar a notação para três dígitos.

Na notação abreviada cada um dos três dígitos é automaticamente dobrado conforme exemplos a seguir:

#FFF	=	#FFFFFF
#CF9	=	#CCFF99
#cde	=	#ccdddee
#49c	=	#4499cc

Dica: <https://www.colorschemer.com/css-color-codes/>

HTML – VIDEOS

Para inserir vídeo em uma página web, basta usar o elemento video:

```
<video controls=" autoplay="true" width="400" height="300">  
  <source src="videos/A.mp4" />  
  <source src="videos/A.wmv" />  
  <source src="videos/A.avi" />  
  <source src="videos/A.mpg" />  
  <p>Não suporta o vídeo</p>  
</video>  
:
```

HTML – AUDIOS

Para inserir áudio em uma página web, basta usar o elemento audio:

```
<audio controls="true" autoplay="true">  
  <source src="musicas/A.mp3" />  
  <source src="musicas/A.mid" />  
  <source src="musicas/A.oga" />  
  <p>Não suporta a música</p>  
</audio>
```

BIBLIOGRADIAS – REFERENCIAS



- **FILHO, Ozeas Vieira Santana.** *Introdução à Internet.*
Editora Senac. São Paulo.
- **BENEDICTO, A. D. Victoriano.** *Aprenda em 24 horas Internet.* Editora Campos, 1998.
- **HAHN, Harley; STOUT, Rick.** *Dominando a Internet.*
Editora Makron Books, 1995.

Desenvolvendo Formulário

FORMULARIO – INTRODUÇÃO

Formulários em HTML5 proporcionam uma melhor experiência para usuários por permitirem que eles sejam mais consistentes entre diferentes sites obtendo um feedback imediato sobre a entrada de dados. Essa experiência também é oferecidas à usuários que possuem scripts desabilitados em seus navegadores

FORMULARIO – DESENVOLVIMENTO

Tag <form>

Representa uma seção de um documento que contém controles interativos que permitem ao usuário submeter informação a um determinado servidor web.

A tag <form> possui:

- ✓ **Id / Name:** identificadores do formulário.
- ✓ **Action:** ação que o formulário terá assim que o botão SUBMIT for pressionado.
- ✓ **Method:** forma de envio de dados do formulário, podendo ser GET ou POST



Nome *

E-mail *

Telefone *

Assunto *

Mensagem *

Enviar Formulário de Contato

FORMULARIO – DESENVOLVIMENTO

Tag <form>

Representa uma seção de um documento que contém controles interativos que permitem ao usuário submeter informação a um determinado servidor web.

```
7<body>
8  <form action="/pagina.html" method="get">
9    Nome: <input type="text" name="nome"><br>
10   Sobrenome: <input type="text" name="sobrenome"><br>
11   <input type="submit" values="Enviar">
12 </form>
13 </body>
```

FORMULARIO – DESENVOLVIMENTO

method=“get”

A requisição irá enviar os dados diretamente no link.

method “post”

A requisição irá enviar os dados encapsulados.

FORMULARIO – DESENVOLVIMENTO

Tag <fieldset> e <legend>

<fieldset></fieldset> : Permite que os elementos de formulário sejam agrupados, possibilitando assim uma maior organização dos dados no desenho do formulário.

<legend></legend> : Receberá um texto que permite identificar o grupo de informações que ali devem ser inseridas

```
<form action="/pagina.html" method="get">
  <fieldset>
    <legend>Dados Pessoais</legend>
    Nome: <input type="text" name="nome"><br>
    Sobrenome: <input type="text" name="sobrenome"><br>
    <input type="submit" values="Enviar">
  </fieldset>
</form>
```

FORMULARIO – DESENVOLVIMENTO

Atributo “type”

Text: campo padrão para a digitação de dados.

Password: campo para a digitação de dados no formato para senhas.

Search: um campo de busca, A aparência e comportamento do campo pode mudar ligeiramente dependendo do agente de usuário (browser).

Email: campo para e-mail com formatação e validação. O agente de usuário pode inclusive promover a integração com sua agenda de contatos.

Url: um endereço web, também com formatação e validação.

FORMULARIO – DESENVOLVIMENTO

Exemplo com email:

```
<form action="/pagina.html" method="get">
  <fieldset>
    <legend>Dados Pessoais</legend><br>
    Nome: <input type="text" name="nome"><br>
    Sobrenome: <input type="text" name="sobrenome"><br>
    E-mail: <input type="email" name="email"><br><br>
    <input type="submit" value="Enviar">
  </fieldset>
</form>
```

FORMULARIO – DESENVOLVIMENTO

Mais sobre o atributo “type”:

Tel: campo para números de telefone.

Date: campo para inserção de datas. Abre um calendário onde o usuário seleciona uma data específica.

Time: campo para inserção da hora. Compreende valores para hora e para minutos.

DateTime-Local: campo para inserção de data e hora.

Month: campo para entrada/seleção do mês.

Week: campo para entrada/seleção da semana.

FORMULARIO – DESENVOLVIMENTO

Mais sobre o atributo “type”:

Number: campo para entrada de números. Podemos definir a faixa de valores com as opções min e max, além de determinar que sejam exibidos números dentro de um intervalo com a opção step.

Exemplo: <input type=“number” min=“5” max=“30” step=“3”>

Range: Renderiza uma barra onde o usuário seleciona uma escala de valores. Possui também as opções de min – max – step.

Mais sobre o atributo “type”:

Radio: botão de seleção única, apresenta várias opções mas o usuário pode apenas escolher, seleciona, uma opção.

Se usar checked, o botão já será selecionado.

CheckBox: botão de seleção múltipla, o usuário pode selecionar quantas opções ele quiser. Se usar checked, os botões já serão selecionados.

File: insere um campo para busca e anexo de documentos.

FORMULARIO – DESENVOLVIMENTO

Mais sobre o atributo “type”:

Button: insere um botão qualquer. Às vezes usamos para chamarmos alguma função em JavaScript.

Submit: envia os dados digitados, assim que este botão é pressionado seu form irá ler o conteúdo do atributo action e direcionar os dados digitados para ele.

Reset: limpa as informações digitadas no formulário.

TextArea: Permite a inserção de uma caixa de texto de múltiplas linhas. A ideia é que o usuário tenha uma área maior para digitação de dados, sendo muito usado para mensagens, comentários, sugestões, etc.

FORMULARIO – DESENVOLVIMENTO

Mais sobre o atributo “type”:

Select/Option: Campo que irá permitir a escolha de uma determinada opção inserida em uma lista acessada por um botão do tipo dropdown. Cada item dessa lista deverá estar contida em um elemento `<option></option>`.

```
<select>
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="mercedes">Mercedes</option>
    <option value="audi">Audi</option>
</select>
```

FORMULARIO – DESENVOLVIMENTO

Outros atributos:

Required: Torna um campo de formulário obrigatório ou seja, seu valor precisa ser preenchido.

Autofocus: Dá foco automático a um determinado campo assim que o navegador carregar a página.

Maxlength: Limita a quantidade de caracteres em um campo de formulário.

FORMULARIO – DESENVOLVIMENTO

Código de exemplo:

```
<form action="/pagina.html" method="post">
  <fieldset>
    <legend>Dados Pessoais</legend><br>
    <div>
      <label> Nome: </label> <input type="text" name="nome" maxlength="20"><br>
    </div>

    <div>
      <label> Sobrenome: </label> <input type="text" name="sobrenome"><br>
    </div>
    <div>
      <label> E-mail: </label><input type="email" name="email"><br><br>
    </div>
    <div>
      <label>Trabalha?</label>
      <select>
        <option value="sim"> Sim </option>
        <option value="nao"> Não </option>
      </select>
    </div>
    <input type="submit" values="Enviar">
  </fieldset>
</form>
```

BIBLIOGRADIAS – REFERENCIAS



- **Curso de HTML5 W3C**
<http://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>
- **Mozilla Developer**
https://developer.mozilla.org/ptBR/docs/Web/Guide/HTML/Forms/Meu_primeiro_formulario_HTML



CSS

CSS - CASCADING STYLE SHEETS

Definição

É uma linguagem de estilos utilizada para definir a apresentação de documentos escritos em HTML.

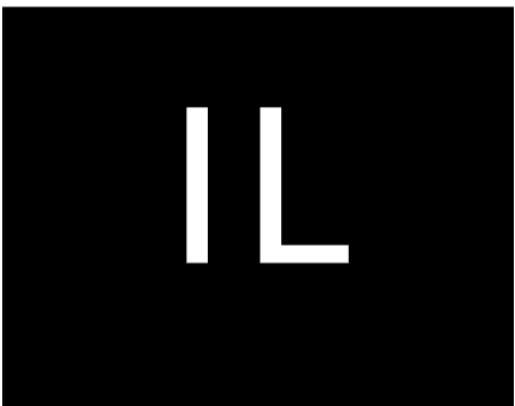
Linguagem criada pelo W3C para definir estilos (cores, tipologia, posicionamento, etc.) em páginas web.

Possibilita que determinadas propriedades sejam aplicadas ao mesmo tempo a todos elementos de uma página ou site que estejam marcados com uma tag específica.

Facilitam a criação, formatação e manutenção de páginas web.

CSS - CASCADING STYLE SHEETS

Tipos de declarações CSS



CSS Inline

As regras de formatação são inseridas diretamente na tag e se aplicam apenas ao elemento marcado.

Não use !!!



CSS Interno

As regras de formatação são inseridas na seção <head> da página e se aplicam ao documento inteiro.

Use com muita moderação !!!



CSS Externo

As regras de formatação são inseridas em um documento separado do HTML e podem ser usadas por várias páginas.

Use à vontade !!!

CSS - CASCADING STYLE SHEETS

CSS InLine

```
<p style="background-color: #900;"> Texto do parágrafo </p>
```

Com esse tipo de declaração você mistura a formatação de um elemento com o seu respectivo conteúdo.

Caso você queira que a mesma regra seja usada por outro elemento, terá de refazer a mesma declaração no elemento desejado.

É **terrível** dar manutenção nesse tipo de código.

CSS - CASCADING STYLE SHEETS

CSS Interno

```
<style type="text/css">  
  
p{  
  
    background-color: #900;  
  
} </style>
```

Declaração feita na seção `<head>` da página.

Todos os elementos da página que forem declarados na regra serão formatados.

A regra é aplicada apenas naquela página que contém a respectiva formatação.

CSS - CASCADING STYLE SHEETS

CSS Interno - Exemplo

```
1 <!DOCTYPE HTML>
2 <html lang="pt-br">
3 <head>
4 <meta charset="UTF-8">
5 <link rel="stylesheet" type="text/css"
href="estilo.css">
6 <title></title>
7 </head>
8 <body>
9
10 </body>
11 </html>
```

CSS - CASCADING STYLE SHEETS

CSS Externo

```
<link rel="stylesheet" type="text/css" href="css/style.css">
```

A linha acima deve ser inserida na seção `<head>` de suas páginas, sua função é indicar onde está o arquivo que contém as regras CSS que devem ser usadas na estilização dos elementos da página.

Uma mesma regra CSS dentro de um arquivo externo, pode formatar simultaneamente quantas páginas o desenvolvedor quiser.

Esse é o padrão que **deve** ser usado.

CSS - CASCADING STYLE SHEETS

CSS Externo - Exemplo

Crie um arquivo chamado: **meuestilo.css**, com o seguinte conteúdo:

```
1 ⊖ p{  
2     color:red;  
3     font-size:14px;  
4     border:2px solid red;  
5     width:200px;  
6 }  
7 body{  
8     background-color: pink;  
9 }
```

CSS - CASCADING STYLE SHEETS

CSS Externo - Exemplo

Agora crie na mesma pasta um arquivo chamado: **TesteCSSExterno.html**, com o seguinte conteúdo:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <link rel="stylesheet" href="meuestilo.css">
5 <meta charset="ISO-8859-1">
6 <title>Teste para CSS Externo</title>
7 </head>
8 <body>
9     <p>CSS Externo</p>
10    <p>Testando meu estilo</p>
11 </body>
12 </html>
```

CSS - CASCADING STYLE SHEETS

Efeito Cascata

O efeito cascata define qual regra será aplicada quando há mais de um estilo especificado para o mesmo elemento HTML. Neste caso ele seguirá o padrão abaixo:

- 1 - Estilo padrão do navegador.
- 2 - Folha de Estilo Externa (referenciada e/ou importada).
- 3 - Folha de Estilo Interna (definida na área de cabeçalho do documento).
- 4 - Folha de Estilo Inline (dentro de um elemento HTML).

CSS - CASCADING STYLE SHEETS

Regra CSS

Uma regra CSS é uma declaração com sintaxe própria que indica como será o estilo de um ou mais elementos HTML. Um conjunto de regras CSS formam uma Folha de Estilos.

É composta por: um seletor, uma propriedade e um valor.

SELETOR

{

PROPRIEDADE: VALOR;

}

CSS - CASCADING STYLE SHEETS

Elementos da Regra CSS

Seletor

Geralmente é o elemento HTML identificado por sua tag, por uma Classe ou um ID. A regra será válida para o elemento especificado.

Exemplo: <p> <h2>
<body> .umaclasse

Propriedade

É aquilo que queremos alterar no seletor, ou seja, aquilo que queremos alterar na aparência do elemento.

Exemplos: cor, largura, altura, margens, etc.

Valor

Representa o novo valor que a propriedade do seletor irá receber.

Exemplos: a nova cor, a nova largura, a nova altura, etc.

CSS - CASCADING STYLE SHEETS

Exemplo de Regras CSS

```
p{  
    color: #336699;  
}
```

No exemplo acima o seletor é a tag <p>, a propriedade que sofrerá alteração será a cor do texto e o novo valor que ela receberá será a cor azul.

Podemos também formatar mais de uma propriedade na mesma regra, para isso usamos:

```
p{  
    font-size: 25px;  
    font-family: calibri, verdana, 'Trebuchet Ms';  
}
```

CSS - CASCADING STYLE SHEETS

Agrupando Seletores

É possível agrupar vários seletores para uma regra CSS, separando-os por vírgula.

No exemplo abaixo, agrupamos todos os elementos cabeçalho. A cor de todos os cabeçalhos será vermelha:

```
h1, h2, h3, h4, h5, h6{  
    color: #990000;  
}
```

CSS - CASCADING STYLE SHEETS

Seletores

Para utilizar ESTILOS especificamente para uma determinada tag é necessário criar SELETORES, para isso podemos fazer de duas maneiras diferentes:

- Seletores de ID: só pode ser usado por uma única tag HTML.
- Seletores de CLASS: podem ser usados por várias tags.

CSS - CASCADING STYLE SHEETS

Seletores de CLASSE

É possível definir um nome criando assim uma classe, onde serão definidas regras CSS. As classes podem ser aplicadas a qualquer elemento HTML e mais ainda, pode-se aplicar estilos diferentes para o mesmo tipo de elemento do HTML, usando classes diferentes para cada um deles. Exemplo:

Declaração de regras CSS:

```
.corum { color:#000000; }  
.cordois { color:#0000FF; }
```

Aplicação no documento HTML:

```
<p class ="corum">Texto </p>  
<h2 class="cordois">Este cabeçalho é azul.</h2>
```

CSS - CASCADING STYLE SHEETS

Seletor de ID

O seletor ID difere do seletor de classe, por ser **ÚNICO**, ou seja, um seletor ID só pode ser aplicado a **UM** e somente **UM** elemento HTML dentro do documento.

Exemplo:

Declaração de regras CSS:

```
#corum { color:#000000; }  
#cordois { color:#0000FF; }
```

Aplicação no documento HTML:

```
<p id ="corum">Texto </p>  
<p id="cordois">Texto azul.</p>
```

CSS - CASCADING STYLE SHEETS

Div

Cria divisões do seu conteúdo em vários blocos, permitindo assim a sua organização e a criação de layouts.

É um container onde podemos armazenar diversos elementos e qualquer tipo de conteúdo, inclusive outras divs.

```
<div class="aula">  
      
    <p> Texto texto texto texto texto texto texto </p>  
</div>
```

CSS - CASCADING STYLE SHEETS

Crie um novo projeto e adicione um arquivo chamado **index.html**



CSS - CASCADING STYLE SHEETS

Revendo as tag's de um HTML5

<!DOCTYPE html> <!--Informa para o Browser, que trata-se de uma página em HTML5 -->

<html lang="pt-br"> <!--Ajuda por exemplo os buscadores, para saber que essa é uma página do Brasil e o idioma é português-->

<head> <!--É o local que podemos dizer que fica a maior parte da inteligência da página -->

<meta charset="utf-8"><!--Resumindo, serve para mostrar no browser todos caracteres do seu texto, como por exemplo acentuação-->

<!--O que estiver no title, será o título que irá aparecer no browser--> <title> Teste HTML 5 </title>

</head>

<body> <!--No corpo da página (body), podemos entender que iremos inserir as tags que queremos que seja exibida no corpo da página-->

<p>Estrutura básica do HTML 5.</p><!-- A tag p, gera um parágrafo, que será exibida no corpo da página --> </body> </html>

CSS - CASCADING STYLE SHEETS

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Exemplo 1</title>
6 </head>
7 <body>
8     <div id="container">
9         <header>
10            Comunicação Visual
11            <nav class="menu">
12                <ul>
13                    <li><a href="#">Home</a></li>
14                    <li><a href="#">A empresa</a></li>
15                    <li><a href="#">Serviços</a></li>
16                    <li><a href="#">Portifolio</a></li>
17                    <li><a href="#">Clientes</a></li>
18                    <li><a href="#">Contato</a></li>
19                </ul>
20            </nav>
21        </header>
22        <main>
23            <article></article>
24            <aside></aside>
25        </main>
26        <footer> </footer>
27    </div>
28 </body>
29 </html>
```

HANDS – ON CONTINUAÇÃO

Vamos inserir uma imagem na tag “`<main>`”

O elemento `<main>` define o conteúdo principal dentro do `<body>` em seu documento ou aplicação. Entende-se como conteúdo principal aquele relacionado diretamente com o tópico central da página ou com a funcionalidade central da aplicação. O mesmo deverá ser único na página. (fonte: <https://developer.mozilla.org>)

```
<main>
  <figure>
    
    <figcaption id="texto"> Hora de mostrar as
      <br> suas habilidades front-end!
    </figcaption>
  </figure>
  <article></article>
  <aside></aside>
</main>
```

HANDS – ON CONTINUAÇÃO

Vamos criar o artigo principal da página, na tag <article>.

```
<article>
  <h1>Front-end</h1>
  <h2>Habilidades de um front-end</h2>
  <p> Um front-end precisa desenvolver habilidades como: criatividade, lógica de programação, e conhecimentos sólidos em design, UX e UI. As ferramentas são várias e o mercado oferece vários frameworks que prometem agilizar o desenvolvimento de páginas, e assim facilitar a vida de um front-end.
</article>
```

HANDS – ON CONTINUAÇÃO

Vamos criar o menu de apoio para o artigo, por meio da tag <aside>

```
<aside>
  <section>
    <h1>Posts relacionados: </h1>
    <nav>
      <ul>
        <li><a href="https://www.w3.org/">Post 1</a></li>
        <li><a href="https://site112.com/tabela-caracteres-html">Post 2</a></li>
        <li><a href="https://www.w3schools.com/html/html_css.asp">Post 3</a></li>
        <li><a href="https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element">Post 4</a></li>
      </ul>
    </nav>
  </section>
</aside>
```

HANDS – ON CONTINUAÇÃO

Vamos finalizar com o rodapé <footer>

```
<footer>
<p> ©, 2020 - Direitos reservados e protegidos por copyright.
</footer>
```

HANDS – ON CONTINUAÇÃO

Eu disse “finalizar”????? Claro que não né? Falta **estilizar** a nossa pagina, vamos contar com uma ajuda externa....

- Crie um arquivo chamado reset.css
- acesse o site <http://meyerweb.com/eric/tools/css/reset/>
- copie o script que apresenta na página para o arquivo “reset.css” que você criou.

Adicione a linha abaixo:

```
4 <meta charset="ISO-8859-1">
5 <link rel="stylesheet" href="css/reset.css">
6 <link rel="stylesheet" href="css/estilo.css">
7 <title>Exemplo 1</title>
```

HANDS – ON CONTINUAÇÃO

Agora vamos criar o arquivo “estilo.css” e montaremos o nosso CSS. Digite o código abaixo:

```
1°#container{ /* seletor de id # */  
2    margin: auto; /* deixaremos a margem automática */  
3    width: 960px; /* este é um tamanho de segurança */  
4 }  
5  
6°header, footer{  
7    display : block; /* Exemplo de elemento de bloco <div>,  
8    poderíamos colocar inline, exemplo de um elemento inline <a>*/  
9    min-height:100px; /* mantém uma altura mínima da tela*/  
10   overflow: hidden; /* oculta o conteúdo se a página ultrapassar a altura */  
11 }
```

HANDS – ON CONTINUAÇÃO

Agora, alteramos no nosso **index.html**:

```
<h1 class="Logo" align="center">Comunicação Visual</h1>
```

E:

```
<nav class="menu"
```

HANDS – ON CONTINUAÇÃO

Continuamos o nosso estilo:

```
19º .Logo { /* seletor de classe */  
20   background-color: red;  
21   background-repeat: no-repeat;  
22   width: 150px; height: 150px;  
23   background-position: center;  
24   background-size: 150px 150px;  
25   display: block; /* Como <a> é inline precisamos colocar block  
26   por que inline não possui altura nem largura */  
27 }
```

HANDS – ON CONTINUAÇÃO

Continuamos o nosso estilo:

```
29 .menu ul li{ /* seletor de classe para as tags ul e li*/
30     width:160px;
31     display:block;
32     float:left;
33     background:#999;
34     text-align:center;
35     line-height:2.5em; /* 2.5 * o tamanho da fonte*/
36     border-bottom:4px solid black;
37 }
```

HANDS – ON CONTINUAÇÃO

Vamos aprimorar ainda mais nosso menu:

```
39° .menu ul li a{  
40    color:inherit; /*permanece o estilo herdado*/  
41    text-decoration:none;  
42    display:block;  
43 }  
44° .menu ul li:hover{  
45    border-color:#666;  
46    background-color:#f7f7f7;  
47    color:#000;  
48    transition:all 1s ease-in;  
49    /*all = todos os elementos  
50    1s = tempo em segundos  
51    ease-in tipo de animação, poderia ser:  
52    step-end  
53    steps (4, end)  
54    linear*/  
55 }
```

HANDS – ON CONTINUAÇÃO

Vamos aprimorar agora o nosso artigo:

```
57°article, aside{  
58    min-height:200px;  
59    padding:20px;  
60 }  
61  
62°article{  
63    background-color:#f1f1f1;  
64    width:678px;  
65    float:left;  
66    margin-right:20px;  
67    border-right:1px solid #ccc;  
68 }
```

HANDS – ON CONTINUAÇÃO

Continuamos a aprimorar o nosso artigo:

```
70oarticle h1{  
71    font-size:32px;  
72    text-transform:uppercase;  
73 }  
74  
75oarticle h2{  
76    font-size:24px;  
77    padding-bottom:15px;  
78 }
```

HANDS – ON CONTINUAÇÃO

Vamos para o `aside` agora:

```
80°aside{  
81    background-color:#d1d1d1;  
82    width:678px;  
83    float:left;  
84 }  
85  
86°aside h1{  
87    font-size:24px;  
88 }
```

HANDS – ON CONTINUAÇÃO

Últimos detalhes do aside:

```
90°aside a{  
91    color:#000;  
92    text-decoration:none;  
93    padding-left: 10px;  
94 }  
95  
96°aside a:hover{  
97    color:#666;  
98    text-decoration:underline;  
99 }
```

HANDS – ON CONTINUAÇÃO

Estilizando o **footer**:

```
101 @footer{  
102     border-top:1px solid #ccc;  
103     margin-top:10px;  
104     font-size:12px;  
105     line-height:40px;  
106     text-align:center;  
107 }
```

HANDS – ON CONTINUAÇÃO

Quase terminando, agora vamos para o **figure figcaption =>**

Provavelmente você precisará mexer nas propriedades de altura e largura da sua imagem, conforme apresentado abaixo:

```
...  
<figure>  
  
```

```
figure{  
  position: relative;  
  height: 250px;  
}  
  
figure figcaption{  
  background: rgba(0,0,0,0.7);  
  color: #fff;  
  font-size: 48px;  
  display: inline;  
  line-height: 1.20em;  
  position: absolute;  
  bottom: 12px;  
  right: 10px;  
  text-align: right;  
  padding: 10px;  
  border: 10px solid #fff;  
}
```

BIBLIOGRADIAS – REFERENCIAS



- **FILHO, Ozeas Vieira Santana.** *Introdução à Internet.*
Editora Senac. São Paulo.
- **BENEDICTO, A. D. Victoriano.** *Aprenda em 24 horas Internet.* Editora Campos, 1998.
- **HAHN, Harley; STOUT, Rick.** *Dominando a Internet.*
Editora Makron Books, 1995.



BOOTSTRAP

O QUE É BOOTSTRAP?

Bootstrap é um framework front-end que facilita a vida dos desenvolvedores web a criar sites com tecnologia mobile (responsivo) sem ter que digitar uma linha de CSS para “fazer e acontecer”. Não é a toa que o termo “Bootstrap” em inglês significa “inicialização”, algo que possui um ponto de partida..

- Além disso, o Bootstrap possui uma diversidade de componentes (plugins) em JavaScript (jQuery) que auxiliam o designer a implementar: tooltip, menu-dropdown, modal, carousel, slideshow, entre outros sem a menor dificuldade, apenas acrescentando algumas configurações no código, sem a necessidade de criar scripts e mais scripts.
- Por exemplo, caso queira implementar um componente de tooltip em seu layout, sem utilizar o Bootstrap, seria necessário:
 - encontrar um plugin que tenha esse comportamento e que funcione com seu tema;
 - acrescentar o script ao html;
 - inicializar o plugin através de script;
 - por fim, criar uma estrutura baseada no plugin.

Bootstrap.css

Bootstrap.css é um framework CSS que organiza e gerencia o layout de um site. Enquanto o HTML gerencia o conteúdo e a estrutura de uma página na web, o CSS lida com o layout dela. Por essa razão, ambas as estruturas precisam coexistir para desempenhar uma ação particular.

Por conta das suas funções, o CSS permite criar um visual uniforme em quantas páginas e do jeito que você quiser.

Diga adeus às horas de edição manual só para mudar a largura de uma borda numa linha.

Com CSS, tudo o que você precisa fazer é referenciar as páginas web no arquivo CSS. Qualquer alteração necessária pode ser feita nesse mesmo arquivo sozinho.

Bootstrap.js

O arquivo Bootstrap.js é a parte central do Bootstrap. Ele consiste em arquivos JavaScript, responsáveis pela interatividade de um site.

Para poupar tempo em sempre terem que escrever sintaxes de JavaScript, os desenvolvedores tendem a usar jQuery. Essa é uma plataforma de código-aberto popular que permite adicionar várias funcionalidades em um site.

Abaixo temos alguns exemplos do que o jQuery pode fazer:

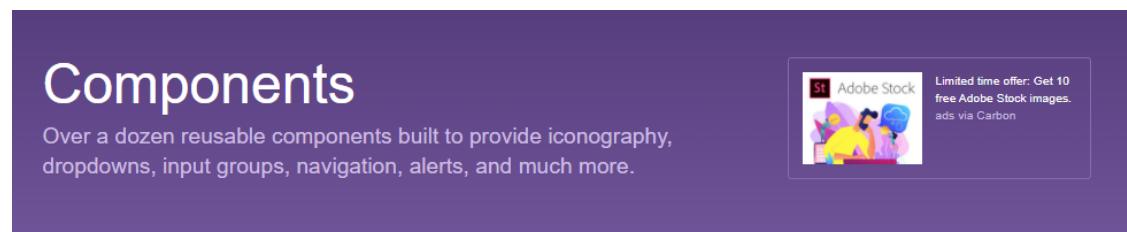
- Executar funções Ajax, como remover dados de outros locais e de forma dinâmica;
- Criar widgets usando uma coleção de plugins JavaScript;
- Criar animações personalizadas usando propriedades do CSS;
- Adicionar dinamismo e interatividade aos conteúdos de um site.

Mesmo que um Bootstrap com propriedades CSS e elementos HTML funcione bem, ele precisa do jQuery para criar um design responsivo. Caso contrário, você vai poder usar apenas as partes estáticas e inativas do CSS.

Glyphicons

Ícones são uma parte integral do front-end de um site. Eles são frequentemente associados com certos tipos de ações e dados disponíveis na interface do usuário. E o Bootstrap usa os Glyphicons para preencher essa necessidade.

O Bootstrap já tem vários símbolos Glyphicons gratuitos já desbloqueados para usar desde o começo. A versão gratuita dele tem um visual padrão, mas dá conta do recado para as funções mais essenciais.



Glyphicons

Available glyphs

Includes over 250 glyphs in font format from the Glyphicon Halflings set. [Glyphicons Halflings](#) are normally not available for free, but their creator has made them available for Bootstrap free of cost. As a thank you, we only ask that you include a link back to [Glyphicons](#) whenever possible.

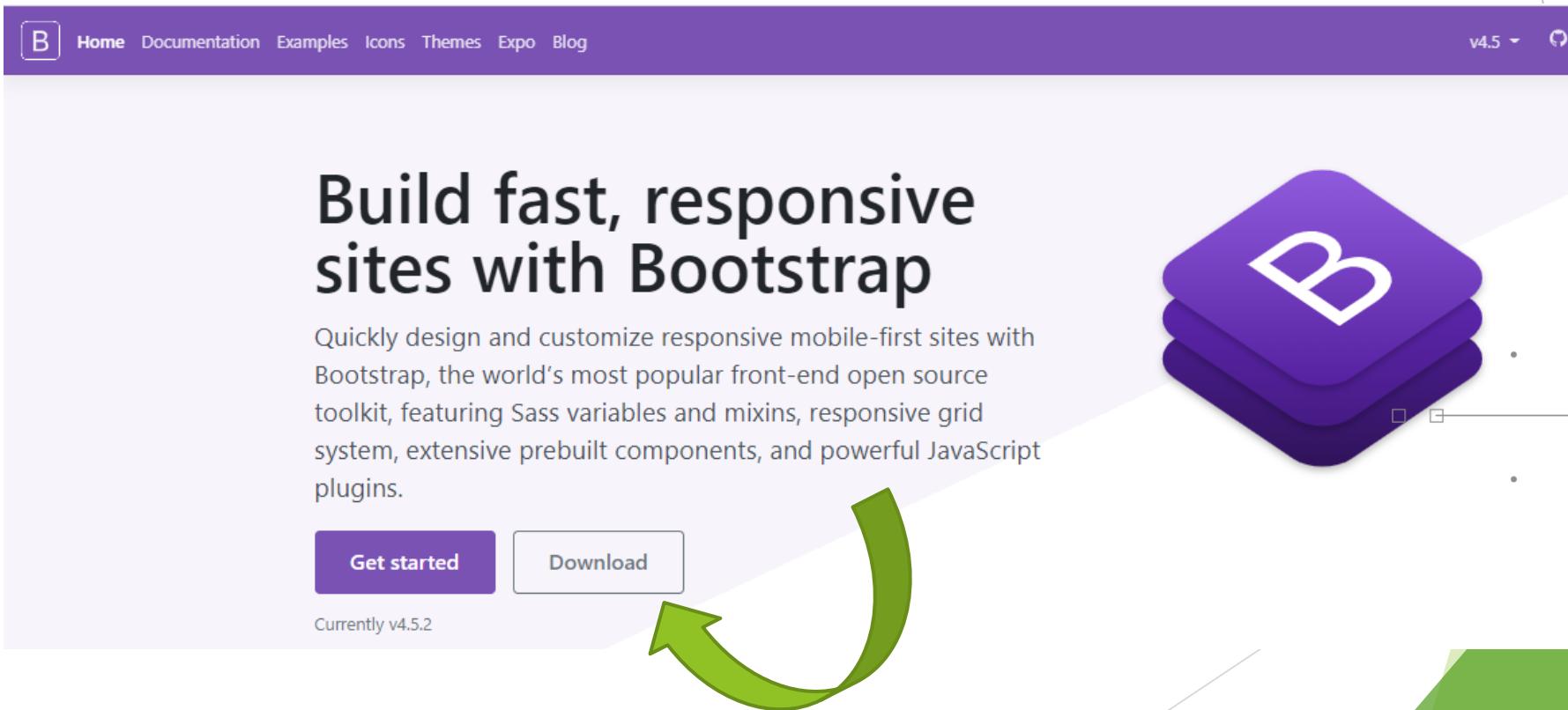
<code>*</code> glyphicon glyphicon- asterisk	<code>+</code> glyphicon glyphicon- plus	<code>€</code> glyphicon glyphicon- euro	<code>€</code> glyphicon glyphicon- eur	<code>-</code> glyphicon glyphicon- minus	<code>☁</code> glyphicon glyphicon- cloud	<code>✉</code> glyphicon glyphicon- envelope	<code>📝</code> glyphicon glyphicon- pencil
<code>↖</code> glyphicon glyphicon- alignleft	<code>.Center</code> glyphicon glyphicon- aligncenter	<code>↗</code> glyphicon glyphicon- alignright	<code>↔</code> glyphicon glyphicon- alignjustify	<code>★</code> glyphicon glyphicon- star	<code>☆</code> glyphicon glyphicon- star	<code>👤</code> glyphicon glyphicon- user	<code>🎞</code> glyphicon glyphicon- film

Search...

- Glyphicons
- Dropdowns
- Button groups
- Button dropdowns
- Input groups
- Navs
- Navbar
- Breadcrumbs
- Pagination
- Labels
- Badges
- Jumbotron
- Page header

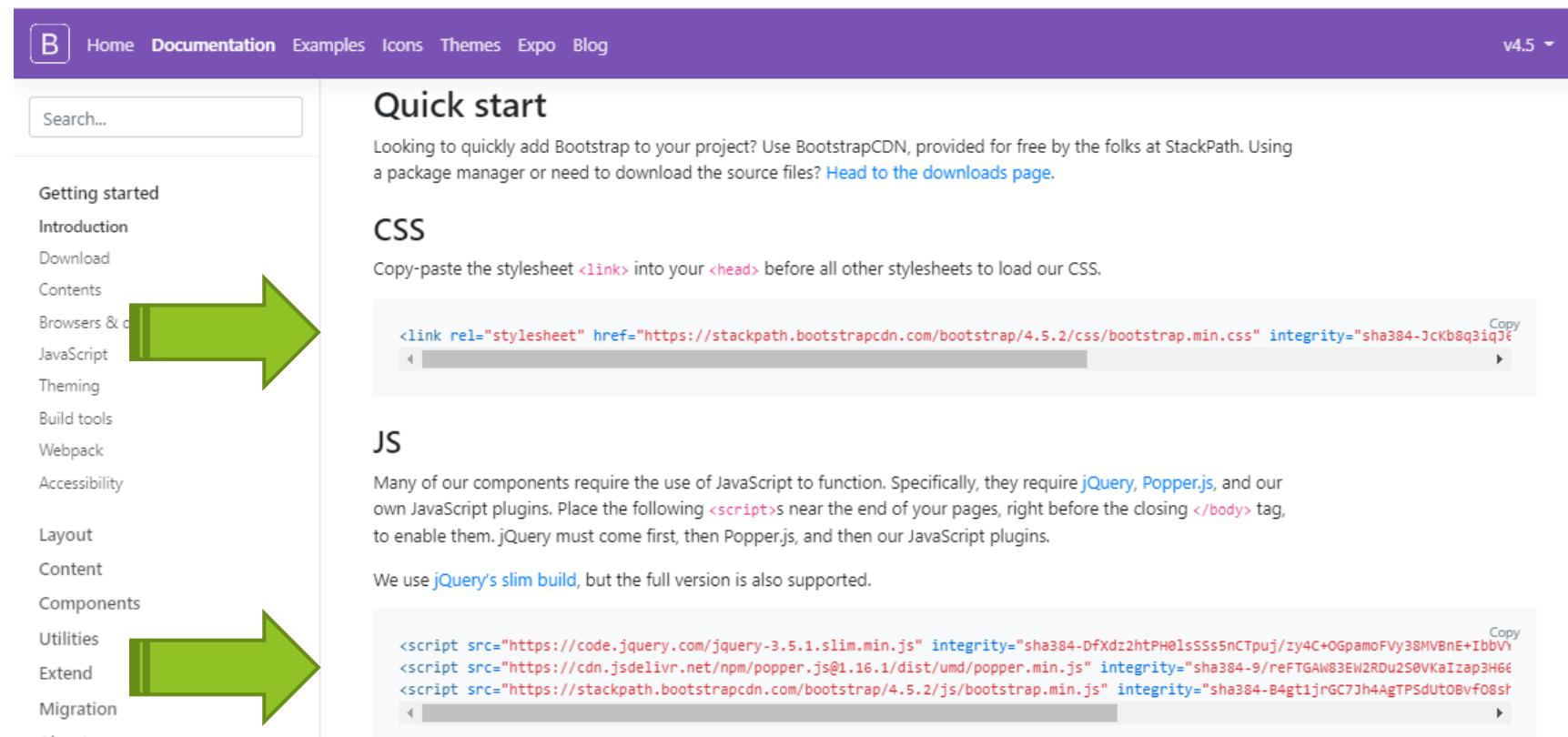
Como inserir um FrameWork BootStrap

Para se configurar um ambiente em BootStrap, deveremos entrar no endereço: <https://getbootstrap.com/>
Em seguida devemos clicar no botão de Get started



Como inserir um FrameWork BootStrap

Copiar os fontes para a nossa página index.html



B Home Documentation Examples Icons Themes Expo Blog v4.5 ▾

Search...

Getting started

- Introduction
- Download
- Contents
- Browsers & c
- JavaScript
- Theming
- Build tools
- Webpack
- Accessibility

Quick start

Looking to quickly add Bootstrap to your project? Use BootstrapCDN, provided for free by the folks at StackPath. Using a package manager or need to download the source files? [Head to the downloads page.](#)

CSS

Copy-paste the stylesheet `<link>` into your `<head>` before all other stylesheets to load our CSS.

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JcKb8q3iqJ6" data-bbox="275 608 755 635" style="border: 1px solid #ccc; padding: 5px; border-radius: 5px; background-color: #f9f9f9;">Copy
```

JS

Many of our components require the use of JavaScript to function. Specifically, they require [jQuery](#), [Popper.js](#), and our own JavaScript plugins. Place the following `<script>`s near the end of your pages, right before the closing `</body>` tag, to enable them. jQuery must come first, then Popper.js, and then our JavaScript plugins.

We use [jQuery's slim build](#), but the full version is also supported.

```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-Dfxz2htPH0ls55nCTpuj/zy4C+OGpamoFVy38MVBN+E+IbbVY" data-bbox="275 848 755 875" style="border: 1px solid #ccc; padding: 5px; border-radius: 5px; background-color: #f9f9f9;">Copy
```

```
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H6" data-bbox="275 875 755 895" style="border: 1px solid #ccc; padding: 5px; border-radius: 5px; background-color: #f9f9f9;">Copy
```

```
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-B4gt1jrGC7Jh4AgTPsDUtOBvf08st" data-bbox="275 895 755 915" style="border: 1px solid #ccc; padding: 5px; border-radius: 5px; background-color: #f9f9f9;">Copy
```

Como inserir um FrameWork BootStrap

Para se fazer a chamada do bootstrap, basta abrir o seu fonte em html e fazer a inserção do fonte como demonstrado

No exemplo abaixo:

```
<html>
<head>
<meta charset="utf-8">
<title> Meu Site em Bootstrap </title>
    
    inserir aqui os arquivo CSS e JS do Bootstrap
</head>
<body>
<div class="container">

</div>
</body>
</html>
```

Como inserir um FrameWork BootStrap

Para se fazer a chamada do bootstrap, basta abrir o seu fonte em html e fazer a inserção do fonte como demonstrado
No exemplo abaixo:

```
<html>
<head>
<meta charset="utf-8">
<title> Meu Site em Bootstrap </title>

<link href ="style.css" rel= "stylesheet"/>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

</head>
<body>
<div class="container">

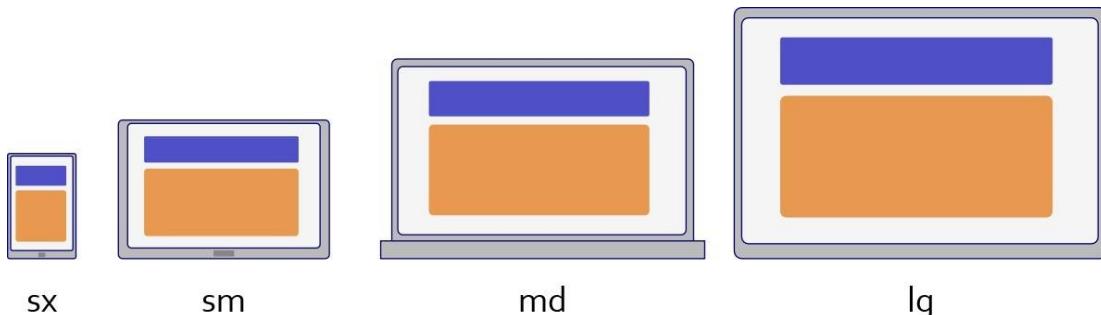
</div>
</body>
</html>
```

Como inserir um FrameWork BootStrap

Bem após inserir o código fonte BootStrap, vamos agora ver os tipos de tamanhos existente os viewports aplicadas nas div's.

Para cada medida de viewport temos um prefixo:

- xs – Extra Small (Extra pequeno)
- sm – Small (Pequeno)
- md – Medium (Médio)
- lg – Large (Grande)



xs	sm	md	lg
Este prefixo será interpretado pelo navegador quando a largura da tela for entre 0 e 767px	Este prefixo será interpretado pelo navegador quando a largura da tela for entre 768px e acima	Este prefixo será interpretado pelo navegador quando a largura da tela for entre 992px e acima	Este prefixo será interpretado pelo navegador quando a largura da tela for entre 1200px e acima
Largura do Container = 100%	Largura do Container = 750px	Largura do Container = 970px	Largura do Container = 1170px
Largura da Coluna = Auto	Largura da Coluna: Aproximadamente 62px	Largura da Coluna: Aproximadamente 81px	Largura da Coluna: Aproximadamente 97px

Como inserir um FrameWork BootSrap

Visualização do layout dos padrões MD,SM E LG

Como inserir um FrameWork BootStrap

Código fonte do exemplo demonstrador anteriormente

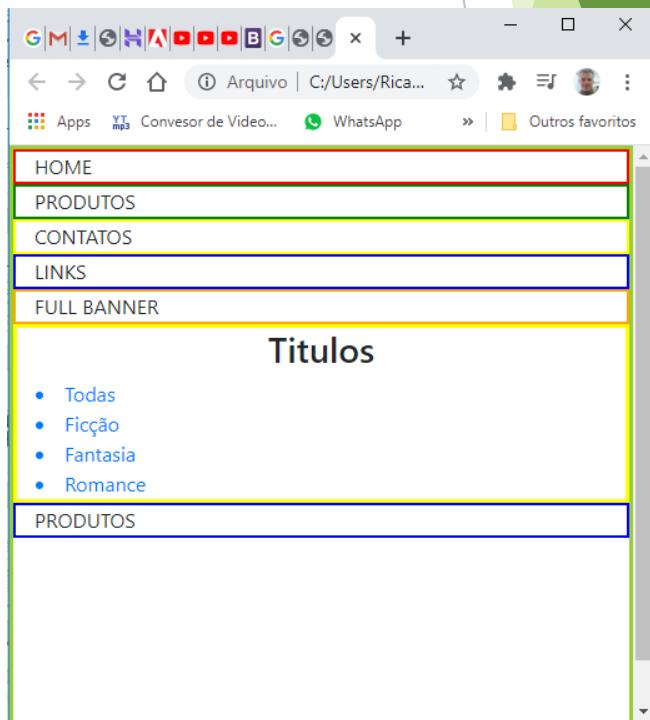
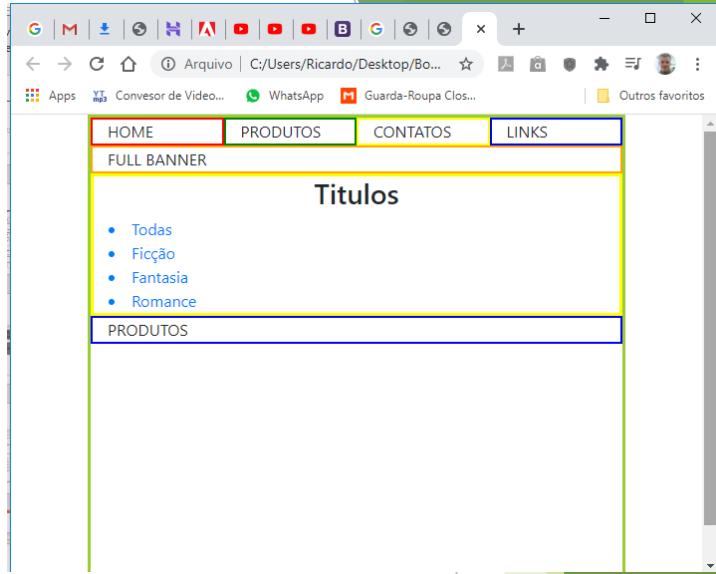
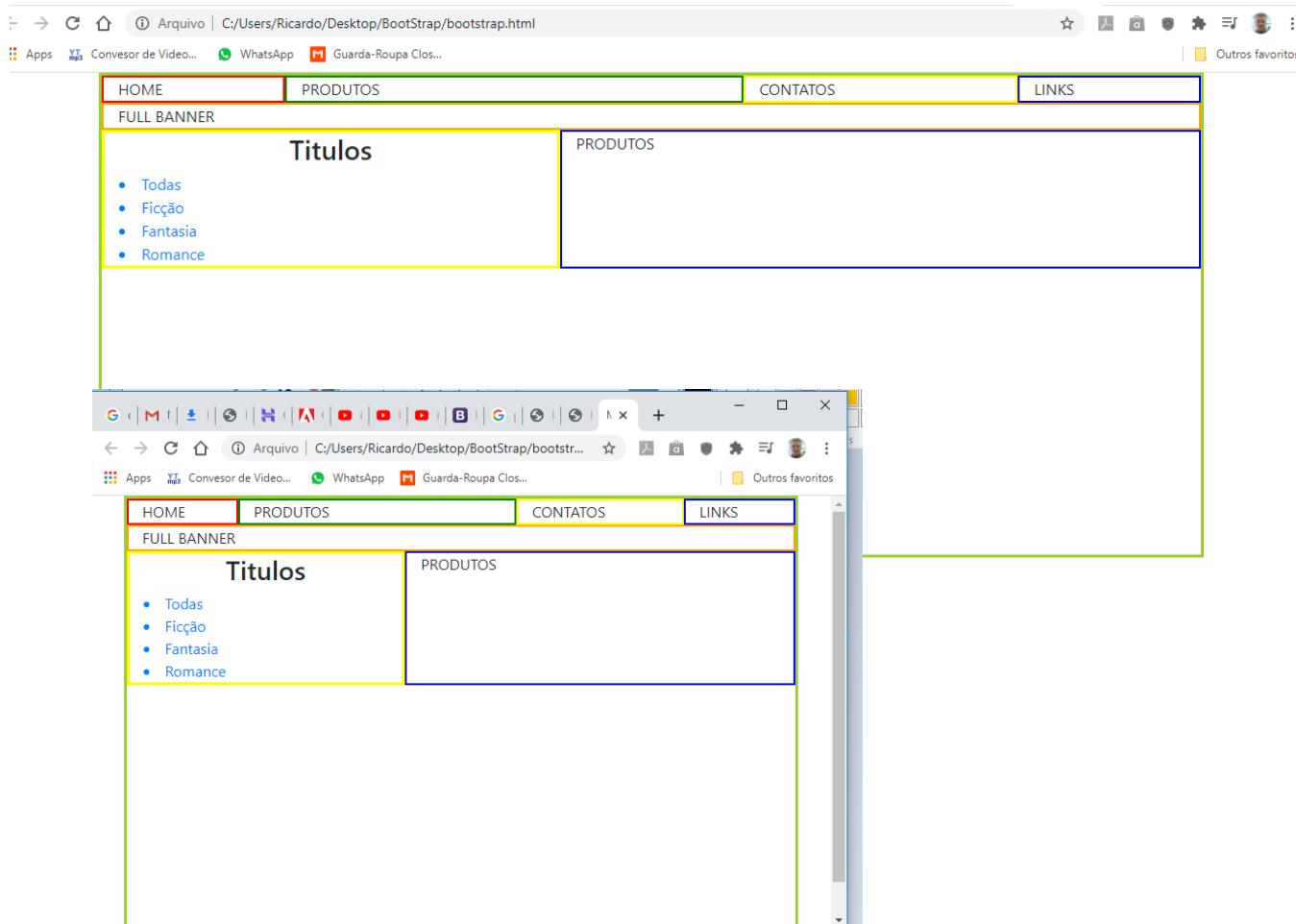
```
<html>
<head>
<meta charset="utf-8">
<title> Meu Site em Bootstrap </title>

<link href ="style.css" rel= "stylesheet"/>
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

</head>
```

```
<body>
<div class="container">
<div class="row">
    <div class="col-md-2 col-sm-3 col-xs-3 div1">HOME</div>
    <div class="col-md-5 col-sm-3 col-xs-3 div2">PRODUTOS</div>
    <div class="col-md-3 col-sm-3 col-xs-3 div3">CONTATOS</div>
    <div class="col-md-2 col-sm-3 col-xs-3 div4">LINKS</div>
</div>
<div class="row">
    <div class="col-md-12 div5">FULL BANNER</div>
</div>
<div class="row">
    <div class="col-md-5 col-xs-6 div6">
        <nav>
            <h3 class="text-center"> Titulos </h3>
            <a href="#" target="_self"><li> Todas</li></a>
            <a href="#" target="_self"><li> Ficção</li></a>
            <a href="#" target="_self"><li> Fantasia</li></a>
            <a href="#" target="_self"><li> Romance</li></a>
        </ul>
    </nav>
</div>
    <div class="col-md-7 col-xs-6 div7">PRODUTOS</div>
</div>
</body>
</html>
```

Resultado da Página em BootStrap nos 4 planos de visualização





JAVASCRIPT

JAVA – SCRIPT

JavaScript é uma linguagem de programação **interpretada**. Foi originalmente implementada como parte dos navegadores web para que scripts pudessem ser executados do lado do cliente e interagissem com o usuário **sem a necessidade** deste script passar pelo servidor.

É atualmente a principal linguagem para programação clientside (lado do cliente) em navegadores web. Foi concebida para ser uma linguagem script com orientação a objetos baseada em protótipos, tipagem fraca e dinâmica e funções de primeira classe. Possui suporte à programação funcional e apresenta recursos como fechamentos e funções.

JAVA SCRIPT – OPERADORES ARITIMÉTICOS

`+` => soma

`-` => subtração

`/` => divisão

`*` => multiplicação

`%` => módulo de uma divisão

`**` => potência

JAVA SCRIPT – OPERADORES LÓGICOS

`&&` => and (e)

`||` => or (ou)

`!` => not (não)

JAVA SCRIPT – OPERADORES RELACIONAIS

`>` => maior

`<` => menor

`>=` => maior ou igual

`<=` => menor ou igual

`!=` => diferente

`==` => igualdade

JAVA SCRIPT – IF

```
if (teste lógico){
```

 condição verdadeira

```
} else {
```

 condição falsa

```
}
```

JAVA SCRIPT – SWITCH

```
switch(expressão){  
  
    case n:  
        bloco de comando;  
        break;  
  
    case n:  
        bloco de comando;  
        break;  
  
    default:  
        bloco de comando;  
  
}
```

JAVA SCRIPT – FOR

```
for (let i = 0 ; i <= 9 ; i++){
```

bloco que será executado...

```
}
```

JAVA SCRIPT – WHILE

```
while (condicionamento) {
```

bloco que será executado... pode ter vários comandos...

```
}
```

JAVA SCRIPT – DO WHILE

```
let i = 0; // variável inicial  
  
do {  
  
    bloco que será executado...  
  
} while (i <= n); // condição
```

JAVA SCRIPT – RECURSOS PARA STRING

Propriedade **length** - retorna o tamanho da string.

Método **toLowerCase()** - retorna a string em letras minúsculas.

Método **toUpperCase()** - retorna a string em letras maiúsculas.

Método **charAt()** - retorna o caracter contido em uma determinada posição.

Método **indexOf('caracter')** - verifica em que posição está determinado caracter, caso não encontre retorna -1. Inicia a busca pelo primeiro caracter.

Método `indexOf('caracter', posição)` - verifica em que posição está o caracter, começando na posição determinada.

Método `lastIndexOf('caracter')` - verifica em que posição está determinado caracter, caso não encontre retorna -1. Inicia a busca pelo último caracter.

Método `substring(início,fim)` - retorna um pedaço da string, você pode determinar o começo e fim deste retorno.

JAVA SCRIPT – ARRAY

Declarando um array

```
var nome =["valor 1", "valor 2", "valor 3", ...];
```

Tamanho do array

```
Nome do array.length;
```

Adicionar um item no fim do array

```
Nome do array.push("conteúdo");
```

JAVA SCRIPT – ARRAY

Remover um item no fim do array

Nome do array.pop();

Adicionar um item no início do array

Nome do array.unshift("conteúdo");

Remover um item no início do array

Nome do array.shift();

JAVA SCRIPT – APLICAÇÃO

Crie um novo arquivo HTML, e digite o código abaixo:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Introdução ao Javascript</title>
6 <script type="text/javascript" src="js/funcoes.js"></script>
7 </head>
8 <body>
9 <form>
10 <input type="button" value="Mostrar nosso OI" onclick="mostrarOi()">
11 </form>
12 </body>
13 </html>
```

JAVA SCRIPT – APLICAÇÃO

Crie um diretório chamado “js” e dentro deste diretório crie um arquivo chamado “funcoes.js”, com o código abaixo:

```
1 function mostrarOi(){
2     alert("Oi");
3 }
```

JAVA SCRIPT - APLICAÇÃO

Volte ao arquivo HTML e adicione as linhas conforme abaixo:

```
1 <!DOCTYPE html>
2<html>
3<head>
4 <meta charset="UTF-8">
5 <title>Introdução ao Javascript</title>
6 <script type="text/javascript" src="js/funcoes.js"></script>
7 </head>
8<body>
9    <p> Recebe um número e retorna se é par ou ímpar
10   <form>
11      <input type="button" value="Mostrar nosso OI" onclick="mostrarOi()">
12      <input type="button" value="Par ou ímpar?" onclick="verificaNumero()">
13   </form>
14 </body>
15 </html>
```

JAVA SCRIPT – APLICAÇÃO

Dentro do arquivo funções (funcoes.js) adicione a nova função, conforme apresentado abaixo:

```
1afunction mostrarOi(){
2    alert("Oi");
3}
4
5bfunction verificaNumero(){
6    var numero=prompt("Digite um número");
7    var resultado = parseInt(numero)%2;
8    if (parseInt(resultado)==0){
9        alert("Número é par");
10    }else{
11        alert("Número é ímpar");
12    }
13 }
```

JAVA SCRIPT – APLICAÇÃO EXEMPLO 2

Crie um novo arquivo HTML, e digite o código abaixo:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Introdução ao Javascript - Exemplo 2</title>
6 <script type="text/javascript" src="js/funcoes.js"></script>
7 </head>
8 <body>
9   <p> Soma de dois valores
10  <form>
11    <label>Digite o primeiro numero: </label>
12    <input type="number" name="numero1"><br>
13    <label>Digite o segundo numero: </label>
14    <input type="number" name="numero2"><br>
15
16    <input type="number" name="resultado" readonly><br>
17
18    <input type="button" value="Exibir resultado"
19      onclick="resultado.value=somarNumeros(numero1.value,numero2.value)">
20  </form>
21 </body>
22 </html>
```

JAVA SCRIPT – APLICAÇÃO EXEMPLO 2

No mesmo arquivo “funcoes.js” acrescente a função abaixo:

```
15 function somarNumeros(num1, num2){  
16     var resultado = parseInt(num1) + parseInt(num2);  
17     return resultado;  
18 }
```

REFERÊNCIAS – BIBLIOGRAFIAS



- **Curso de HTML5 W3C**
<http://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>
- **Mozilla Developer**
https://developer.mozilla.org/ptBR/docs/Web/Guide/HTML/Forms/Meu_primeiro_formulario_HTML

Eu Prof. Ms. Ricardo Alexandre Bontempo
agradeço imensamente ao professor Humberto Delgado pelo consentimento da utilização
do seu material e implementação autorizada para divulgação no curso.

Copyright © 2020 | Professor (a) Prof. Humberto Delgado de Sousa e Prof. Ricardo Alexandre Bontempo

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem
consentimento formal, por escrito, do professor/autor.