

TECNOLOGIA PARA FRONT- END I

HTML e CSS

Facilitador Msc. Angelo F. Dias Gonçalves

Tópicos Teóricos Principais

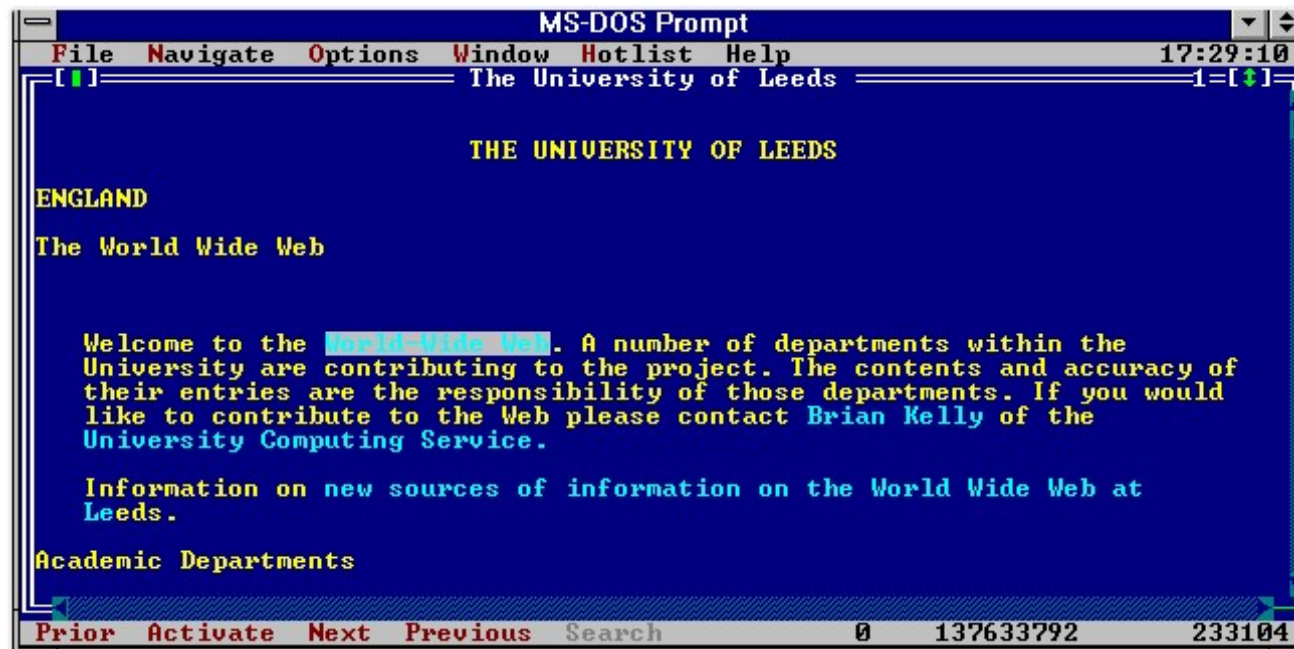
- Evolução dos Navegadores
- Evolução das Aplicações
- Arquitetura Cliente Servidor
- Ferramentas para o Desenvolvimento



Fonte: Google Imagens

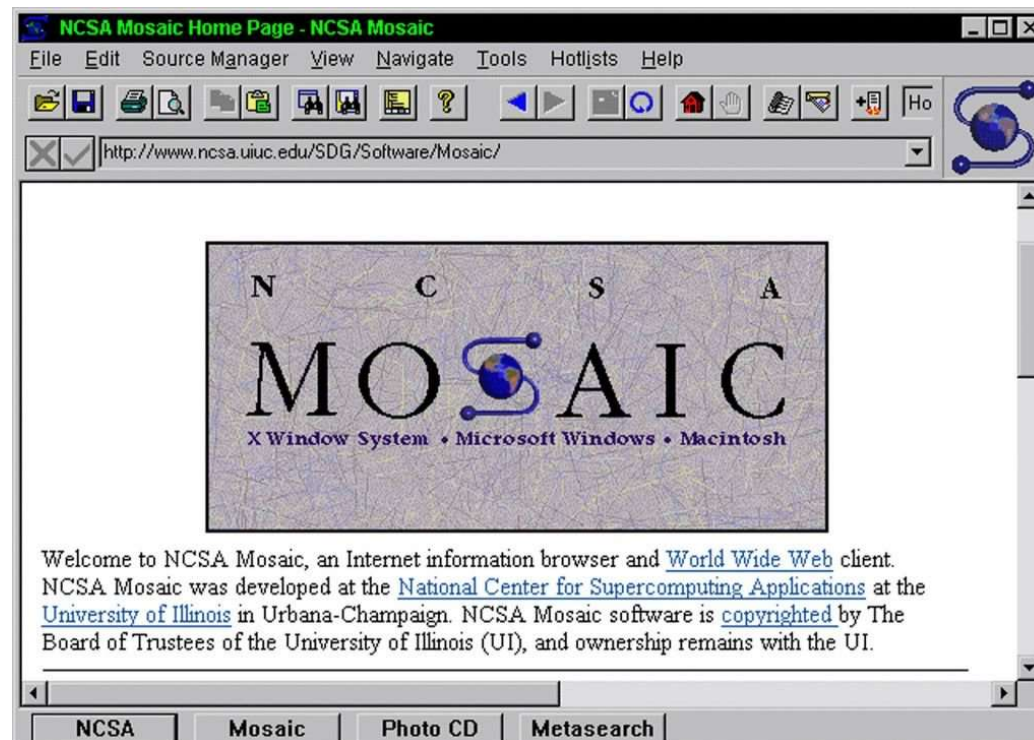
Evolução dos Navegadores

- No começo tudo era acessado pelo terminal
- Tim Beners-Lee criou o primeiro navegador, chamado WorldWideWeb (1990)



Evolução dos Navegadores

- Popularização da Web
- NCSA Mosaic – Navegador Gráfico (1993)



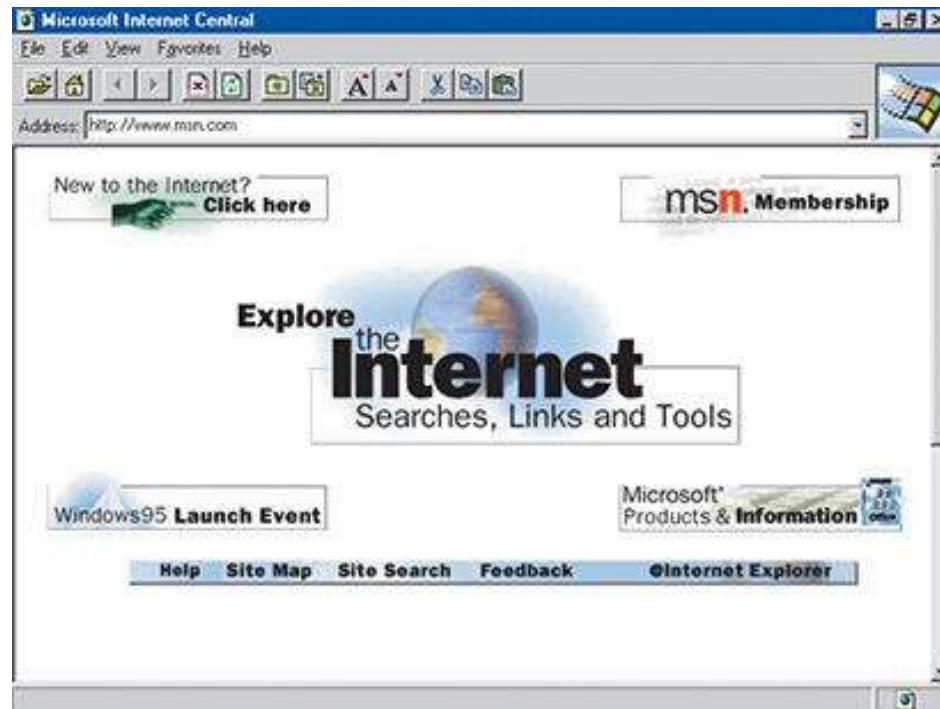
Evolução dos Navegadores

- NetScape
- Primeira Versão Beta do navegador Mozilla 0.96b (1994)



Evolução dos Navegadores

- A Microsoft visualizando o avanço da Web desenvolveu o Microsoft Internet Explorer 1.0 (1995)





Evolução dos
Navegadores





Home has a new hub

The Family Hub™ is a revolutionary new refrigerator with a Wifi enabled touchscreen that lets you manage your groceries, connect with your family and entertain like never before.



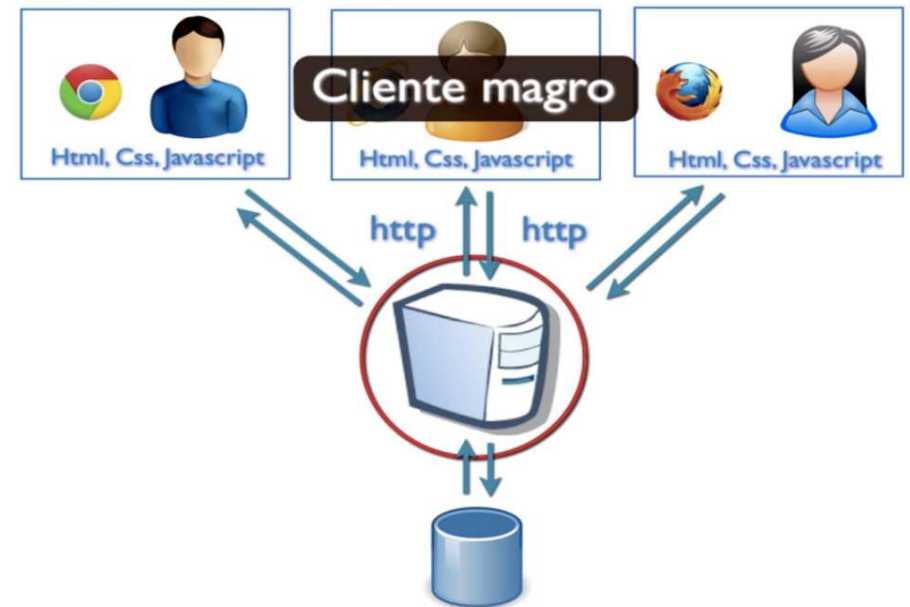
Dispositivos com Browser

Evolução das Aplicações

Aplicação Desktop?



Aplicação na Web?

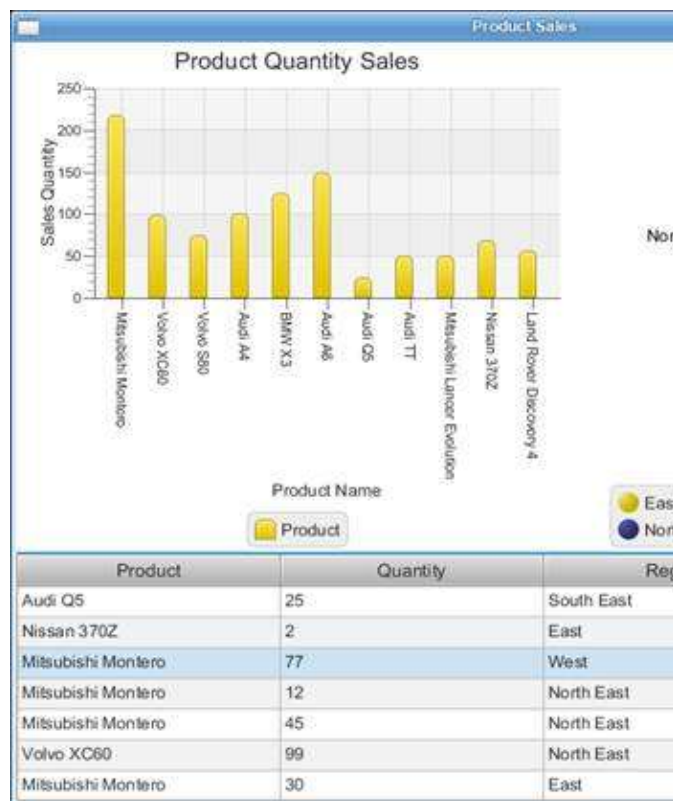


Evolução das Aplicações

Desktop	Web
Componentes ricos	Html. Css. js. e bibliotecas
Orientado a evento	Orientado a request e response
Mantém o estado	Sem estado
Dificulta a manutenção e gerenciamento	Facilita a manutenção e gerenciamento

Evolução das Aplicações

- Componentes Ricos?



SEM CSS



[David Arty Google](#)

Olá,

Sou David, fundador do blog Chief of Design.

Trabalho com design, principalmente com design para web, desde 2009.

Procuro transformar ideias loucas e complexas em peças simples, atrativas e funcionais.

[Saiba Mais](#)

COM CSS



♦ David Arty ♦

Olá,

Sou David, fundador do blog Chief of Design.

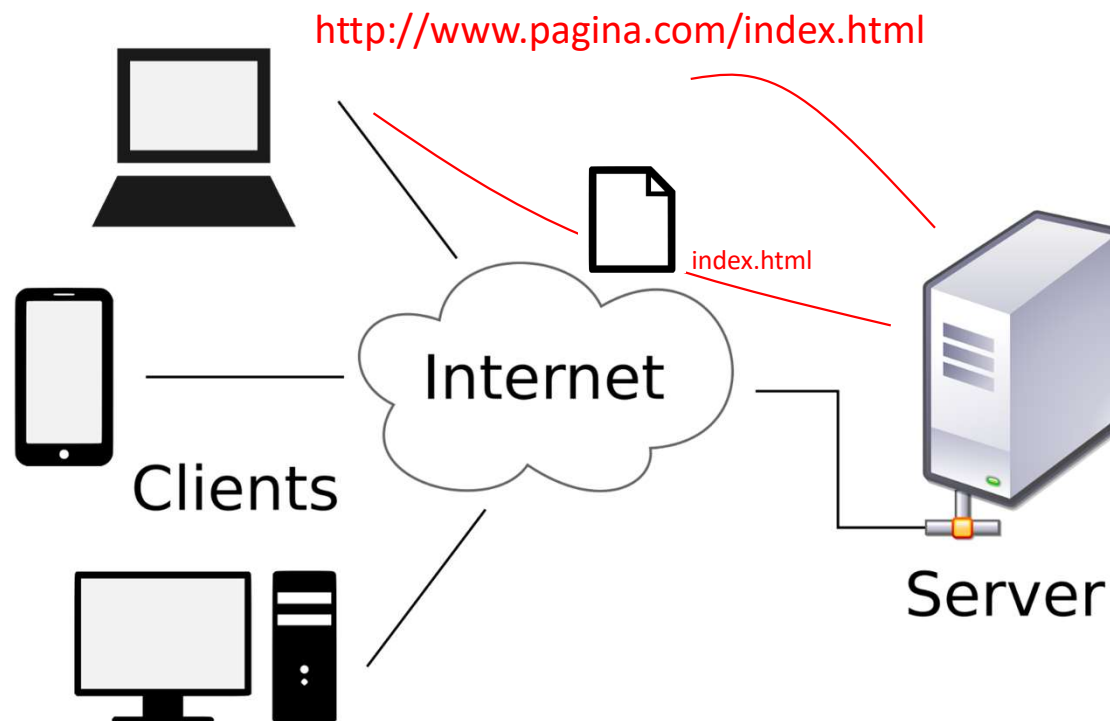
Trabalho com design, principalmente com design para web, desde 2009.

Procuro transformar ideias loucas e complexas em peças simples, atrativas e funcionais.

Saiba Mais

Arquitetura Cliente Servidor

- A internet é uma rede de computadores e obtemos os arquivos através da Web



Protocolo HTTP

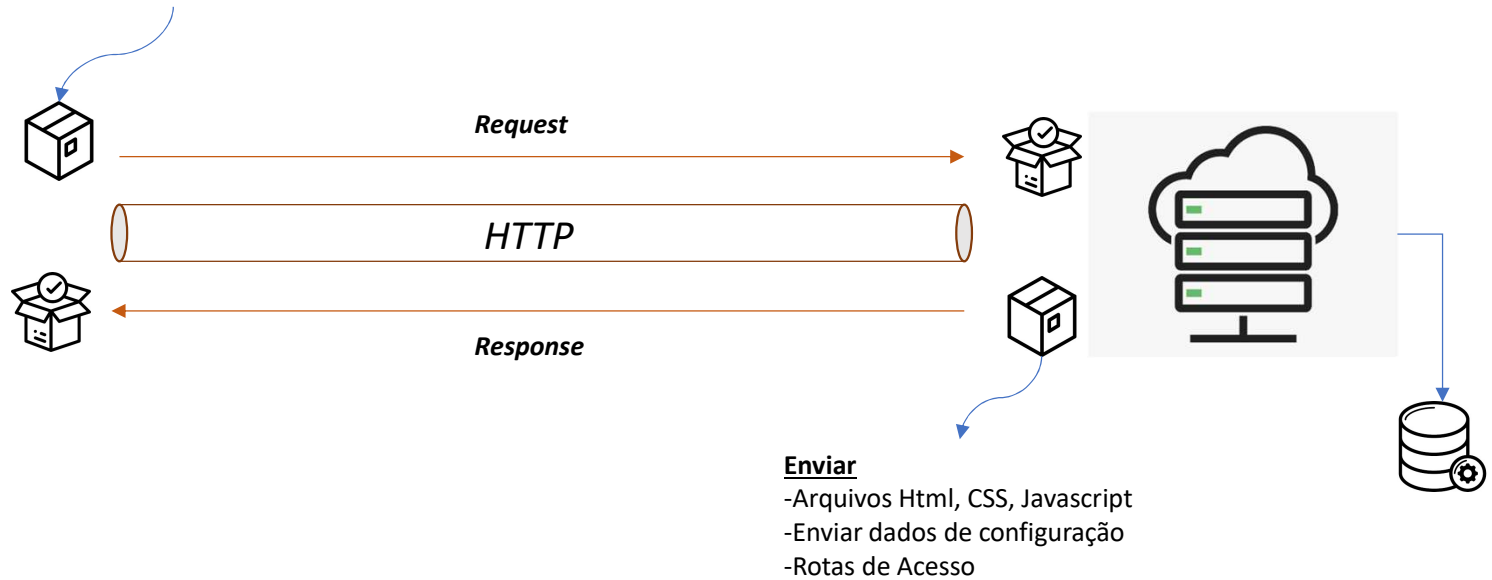
- **Protocolo** é o conjunto de regras sobre o modo como se dará a comunicação entre as partes envolvidas.
- O Hypertext Transfer Protocol, sigla **HTTP** (em português **Protocolo** de Transferência de Hipertexto) é um **protocolo** de comunicação utilizado para sistemas de informação de hipermídia, distribuídos e colaborativos. Ele é a base para a comunicação de dados da World Wide Web.

WWW



Solicitar

- Arquivos Html, CSS, Javascript
- Enviar dados de configuração
- Rotas de Acesso



Ferramentas para o Desenvolvimento



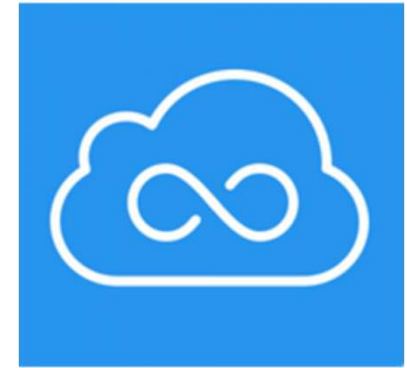
Dreamweaver



Sublime



Notepad



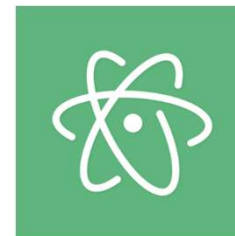
Fiddle



Brackets

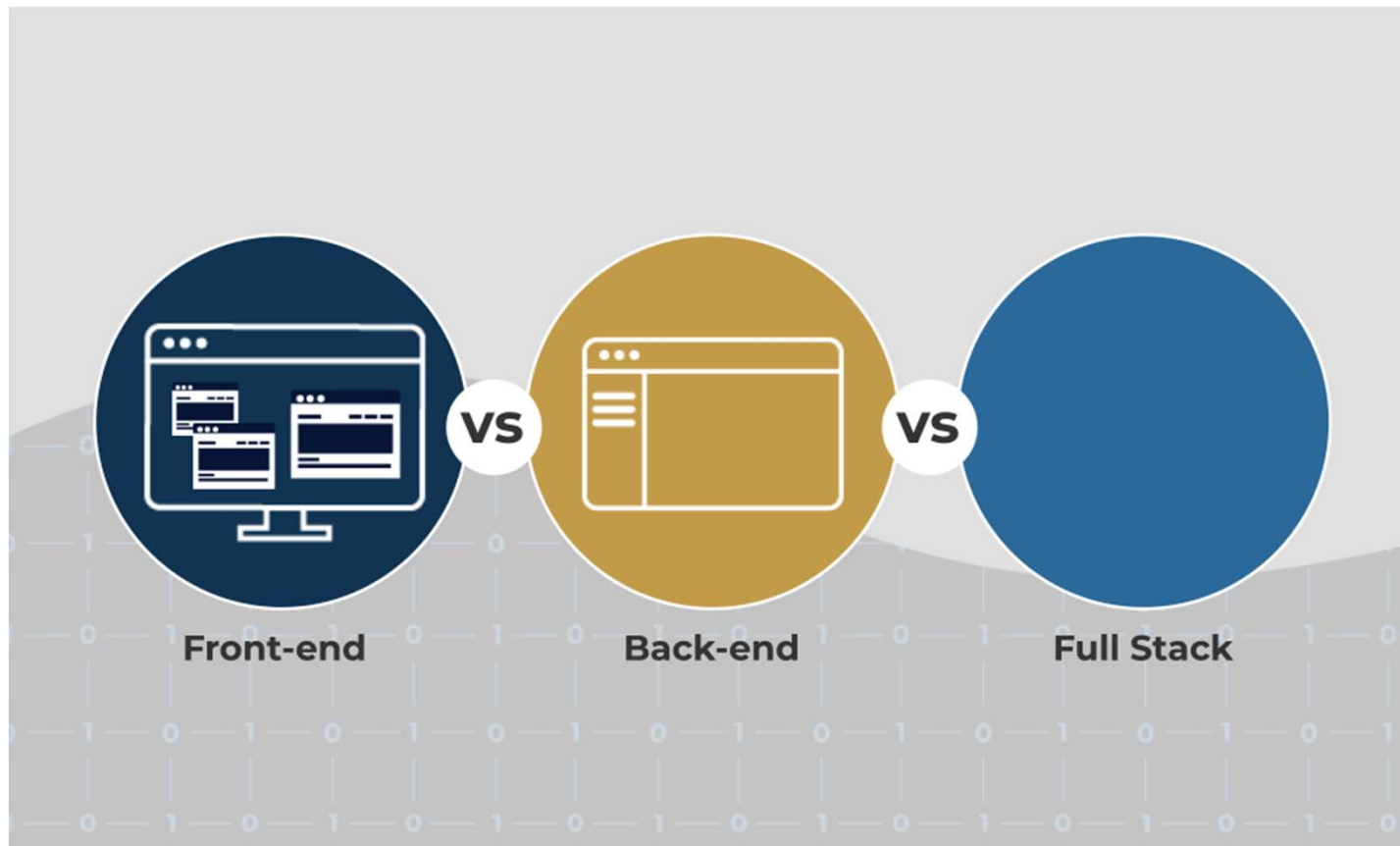


Visual Studio Code

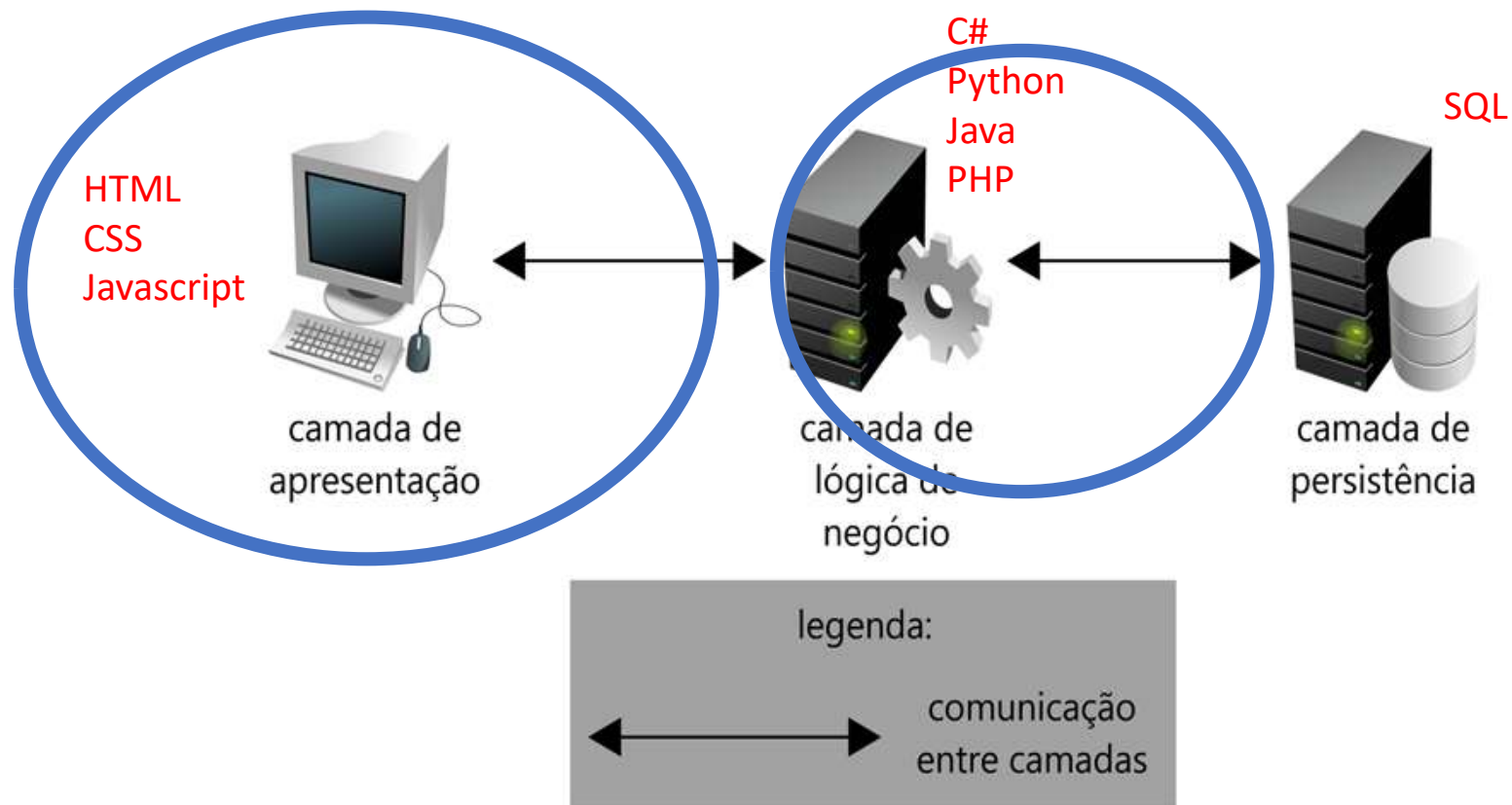



Atom

WWW




Em qual camada a linguagem Javascript e PHP se Enquadra?



Olá, Visitante
Entre Cadastre-se

Buscar

Wow !



Os produtos que você curte. O melhor serviço da internet.

Navegue pelas lojas

Meu Cadastro

Para ser cliente (pessoa Física ou Jurídica), é necessário preencher constantemente o formulário abaixo com os respectivos dados cadastrais. Os campos em **NEGRITO** são de preenchimento obrigatório e essenciais para processarmos o envio do seu futuro pedido.

Pessoa Física

Pessoa Física

*Como gostaria de ser chamado?

* nome completo

* CPF

* data de Nascimento

* sexo

* telefone (DDD - TELEFONE)

telefone celular (DDD - TELEFONE)

* e-mail

* senha Deve ter entre 6 e 8 caracteres

* confirma senha Deve ter entre 6 e 8 caracteres

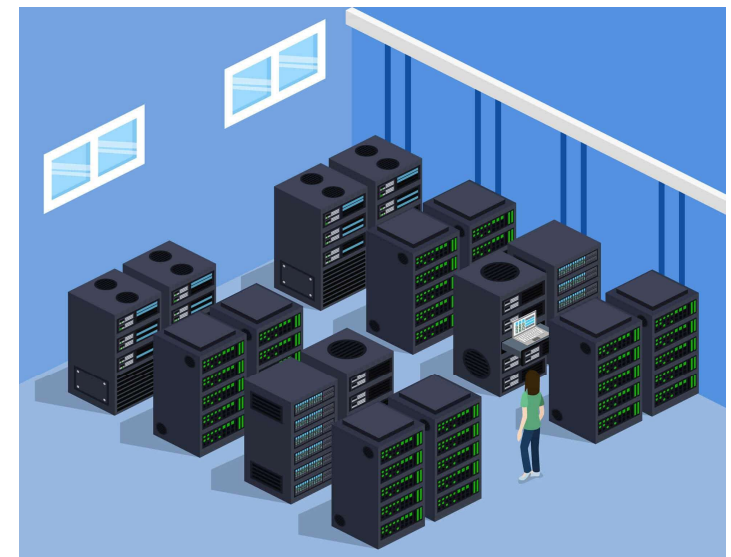
* Receber ofertas do Submarino por e-mail? ☒ Sim ☐ Não

* Receber ofertas do Submarino pelo celular? ☒ Sim ☐ Não

* Receber ofertas do Outlet SouBarato.com.br por e-mail? ☒ Sim ☐ Não

Meu Endereço

Apresentação



Servidor/Banco de Dados

Introdução ao HTML

- Entre todos os recursos envolvidos e necessários para criar uma aplicação web, o HTML é sem dúvida o mais importante.
- A essência de qualquer página web começa no HTML
- O HTML irá informar ao navegador o que cada elemento dentro da página representa, e sem ele não é possível saber o que cada coisa é em uma página web.

Evolução da HTML

- Desde a invenção da web, a HTML evoluiu por oito versões que são:
 - HTML
 - HTML +
 - HTML 2.0
 - HTML 3.0
 - HTML 3.2
 - HTML 4.0
 - HTML 4.01
 - HTML 5

Introdução ao HTML

- Desenvolvida em 1989 por **Tim Beners-Lee** e seus colaboradores no CERN (Conselho europeu de pesquisas nucleares)
 - Manter os documentos Organizados
- Em 1990 foi definida em especificações formais;
- Em 2000 a linguagem tornou-se uma norma internacional
 - Versão Atual HTML5

Introdução ao HTML

- O que significa HTML?
 - **Hypertext Markup Language**
 - Estruturar Documentos
- O que pode conter?
 - Texto, imagens, Figuras, Folhas de Estilos, Cores...
- Quem padroniza tudo isso?
 - W3C (World Wide Web Consortium)
 - Estabelece compatibilidade das informações na Internet

Primeiro Site HTML

<http://info.cern.ch/hypertext/WWW/TheProject.html>

Introdução ao HTML

- HTML
 - Estruturar o conteúdo do documento
- CSS
 - Uma *linguagem de folhas de estilos*. Isso quer dizer que o CSS permite que você aplique estilos seletivamente a elementos em documentos HTML.
- Javascript
 - Fornece mais interação à página web
 - Serve para manipular o comportamento da página

Introdução ao HTML

- Para se comunicar com o navegador é necessário entender a sua linguagem.
- Uma vez que você saiba se comunicar com o navegador, ele saberá apresentar ao usuário a página web.
- **Lembre-se**
 - O HTML não é uma linguagem de programação, na verdade ela é uma **linguagem de marcação**.

Estrutura

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
  </head>
```

```
  <body>
```

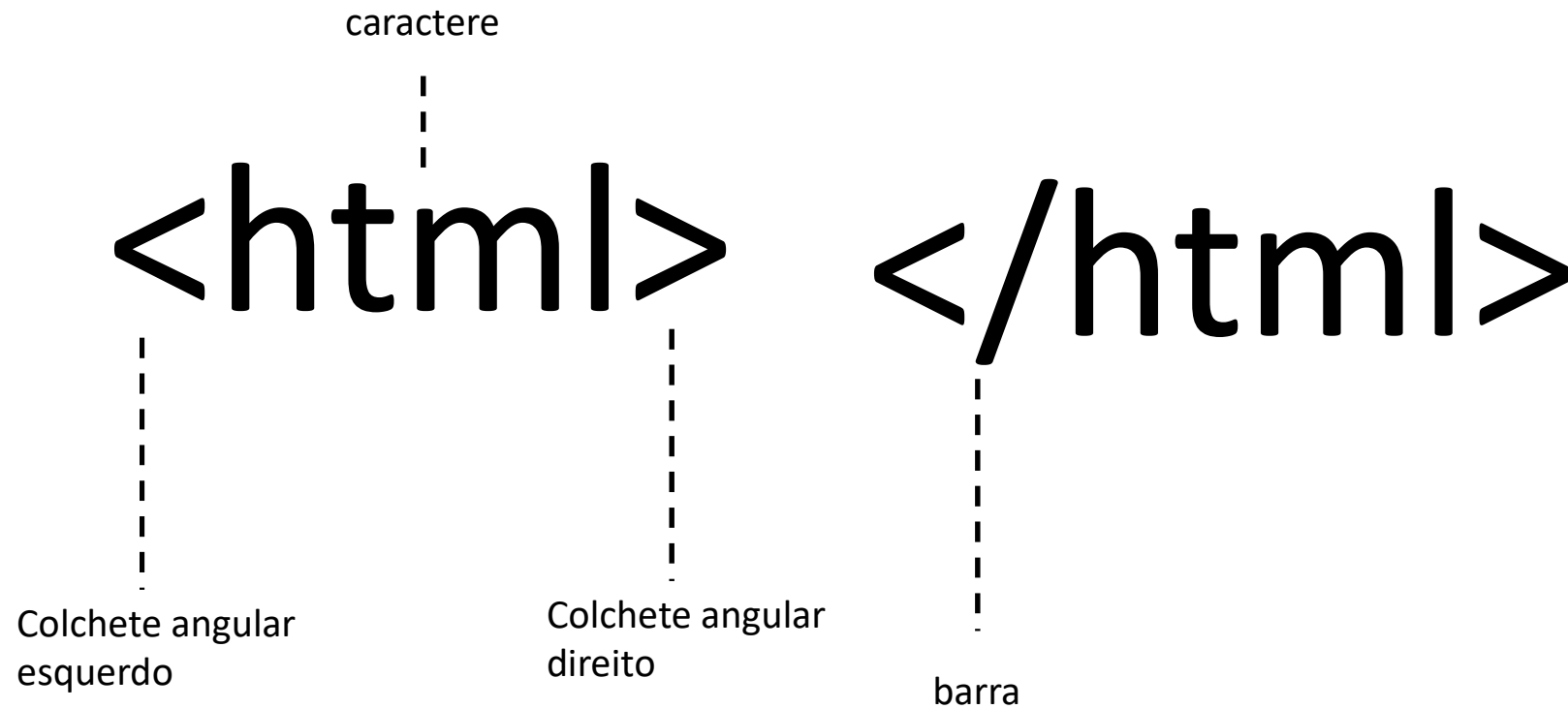
```
    </body>
```

```
</html>
```

Tag (composta < e >)

- **<html>** - Delimitar toda a página.
(todo o código html deve ser escrito dentro dessa tag)

Tags



Estrutura

```
<html>
  <head>
    <title></title>
  </head>
  <body>

  </body>
</html>
```

- **<body>** - Inserir o elementos visuais nas páginas. (texto, imagens, tabelas).

Estrutura

```
<html>
  <head>
    <title></title>
  </head>
  <body>

  </body>
</html>
```

- **<head>** - Configurar outras coisas que não seja conteúdo da página. (Título, encoding, importar códigos javascript e css).
-Aqui estão as definições de lógica sobre o documento

Estrutura

```
<html>
  <head>
    <title></title>
  </head>
  <body>

  </body>
</html>
```

- **<title>** - Inserir um título na janela ou aba da página.

Cabeçalhos

Headings (Cabeçalhos)

- `<h1></h1>`
- `<h2></h2>`
- `<h3></h3>`
- `<h4></h4>`
- `<h5></h5>`
- `<h6></h6>`



Ordem de mais importante
para menos importante.

Parágrafo

- `<p> </p>`

`<p>` - Criar um parágrafo em uma página html.

- Encode (Charset)
 - URL encoding converte caracteres em um formato que pode ser transmitido pela internet.
 - `<meta charset="utf-8"/>`

Criando o 1ª documento HTML

```
<html>
<body>

  <h1>Alô Mundo! ( Hello World! )</h1>
  <h1>Meu primeiro título</h1>
  <p>Meu primeiro título</p>

</body>
</html>
```

- Digite o exemplo acima em um editor de texto simples, como o notepad, gedit, Notepad++, Sublime, **VS Code**;
- Salve o arquivo com a extensão .html

Negrito

- ``
 - Colocando as palavras entre as tags `` podemos exibir os caracteres em negrito.

```
<p>Etiam posuere quam <b>ac quam.</b>  
</p>  
<p><b>Nulla</b> non arcu lacinia neque  
faucibus fringilla. <b>Nulla</b> non lectus  
sed nisl molestie malesuada. Proin in  
tellus sit amet nibh dignissim sagittis.  
Vivamus luctus egestas leo. Maecenas  
sollicitudin.</p>
```



Etiam posuere quam **ac quam.**

Nulla non arcu lacinia neque faucibus fringilla. **Nulla** non lectus sed nisl molestie malesuada. Proin in tellus sit amet nibh dignissim sagittis. Vivamus luctus egestas leo. Maecenas sollicitudin.

Itálico

- `<i>`
 - Colocando as palavras entre as tags `<i></i>` podemos exibir os caracteres em itálico.

```
<p>O <i>Windows</i> 10 é um ótimo SO  
(Sistema Operacional)</b>  
</p>
```

```
<p>O <i>Joystick, mouse e Keyboard</i>  
são dispositivos de entrada em jogos  
digitais</p>
```

O *Windows* 10 é um ótimo SO (Sistema Operacional)

O *Joystick, mouse e Keyboard* são dispositivos de entrada em jogos digitais

Quebras de Linhas

- `
`

- Como já foi visto, o navegador exibirá automaticamente cada novo paragrafo ou título em uma nova linha. Mas se quiser adicionar uma nova quebra de linha no meio de um paragrafo?

```
<p>Ontem,<br /> na aula de Química eu  
aprendi sobre a elemento <br /> H<sub>2  
</sub>O</p>
```

Ontem,
na aula de Química eu aprendi sobre a elemento
H₂O

Linhas Horizontais

- `<hr />`

- Serve para criar uma quebra de linha entre os temas – como uma alteração no tema em um livro ou uma nova cena em uma peça.

```
<p>Ontem na aula de Física eu aprendi  
sobre a fórmula  $E=MC^2$ </p>
```

```
<hr />
```

```
<p>Ontem na aula de Química eu aprendi  
sobre a elemento  $H_2O$ </p>
```

Ontem na aula de Física eu aprendi sobre a fórmula $E=MC^2$

Ontem na aula de Química eu aprendi sobre a elemento H_2O

Formatações (em parágrafos)

- <p>
- Texto em negrito

- <i>Texto em itálico</i>

- <u>Texto sublinhado</u>

- _{Texto subscrito}

- ^{Texto sobrescrito}

- <big>Texto com fonte maior do que o padrão</big>

- <small>Texto com fonte menor do que o padrão</small>

- Texto em itálico

- Texto em negrito
- </p>

Estilos (Spoiler)

- Estilos permitem que sua página tenha diversos tipos de formatações. Eles podem ser declarados com a tag `<style>`.
- Esse exemplo aplica a cor vermelha a todos os elementos do tipo parágrafo:

```
<style>  
    p { color: red; }  
</style>
```

Estilos

- O plano de fundo da página seja alterado para a cor vermelha:

```
<style>
```

```
    body { background-color: red; }
```

```
</style>
```

Estilos

- O plano de fundo da página seja alterado para a cor vermelha:

```
<style>
```

```
    body { background-color: red; }
```

```
</style>
```

Estilos

```
<html>
<head>
  <title>Meus estilos</title>
  <style>
    body {
      background-color: red;
      color:green;
    }
  </style>
</head>
<body>
  <p>Olá mundo</p>
</body>
</html>
```

Listas

Há muitas ocasiões em que precisamos usar listas. A HTML nos fornece três tipos diferentes.

- **Listas Ordenadas:** são listas onde cada item da lista é numerado. **Ex:** Conjuntos de passos para uma receita.
- **Listas não Ordenadas:** são listas que começam com um marcador (em vez de caracteres que indiquem ordem).

Listas

Listas Ordenadas

- ``
 - A lista ordenada é criada com o elemento ``
- ``
 - Cada item da lista é colocado entre uma tag de abertura `` e uma tag de fechamento `` (li significa item da lista).

```
<h1>Bolo - Modo de Preparo</h1>
```

```
<ol>
```

```
  <li>Bata as claras em neve e reserve.</li>
```

```
  <li>Misture as gemas, a margarina e o açúcar até  
  obter uma massa homogênea.</li>
```

```
  <li>Acrescente o leite e a farinha de trigo  
  aos poucos sem parar de bater.</li>
```

```
  <li>...</li>
```

```
</ol>
```

Bolo - Modo de Preparo

1. Bata as claras em neve e reserve.
2. Misture as gemas, a margarina e o açúcar até obter uma massa homogênea.
3. Acrescente o leite e a farinha de trigo aos poucos sem parar de bater.
4. ...

Listas

- Listas Ordenadas

- Utilizando o atributo `type=""` na tag `` é possível indicar o tipo de ordenação.
- `type="1"` – Lista com ordenação numérica
- `type="A"` – Lista com ordenação alfabética Maiúscula
- `type="a"` - Lista com ordenação alfabética Minúscula
- `type="I"` – Lista com ordenação Romana Maiúscula
- `type="i"` - Lista com ordenação Romana Minúscula

Listas

Listas não Ordenadas

- ``
 - A lista não ordenada é criada com o elemento ``
- ``
 - Cada item da lista é colocado entre uma tag de abertura `` e uma tag de fechamento `` (li significa item da lista).

```
<h1>Bolo - Ingredientes</h1>
<ul>
  <li>2 xícaras (chá) de açúcar</li>
  <li>3 xícaras (chá) de farinha de trigo</li>
  <li>4 colheres (sopa) de margarina</li>
  <li>...</li>
</ul>
```

Bolo - Ingredientes

- 2 xícaras (chá) de açúcar
- 3 xícaras (chá) de farinha de trigo
- 4 colheres (sopa) de margarina
- ...

Links

- Criando Links
 - Links são a característica definidora da Web, porque permitem pular de uma página web para outra – possibilitando a própria ideia de navegar ou surfar na internet.
- Os tipos mais comuns de links
 - Links de um site para outro
 - Links de uma página para outra no mesmo site*
 - Links de uma parte da página web para outra parte da mesma página.
 - Links que abrem uma nova página do navegador
 - Links que iniciam o programa de e-mail e endereços um novo e-mail para alguém

Links

- Criados usando o elemento `<a>`
- Os usuários podem clicar qualquer coisa entre a tag de abertura `<a>` e a tag de fechamento ``.
- Você especifica a página de destino do link utilizando o atributo `href`.

Links

Este é a página para a qual o link leva você

Texto em que o usuário clica

`IMDB`

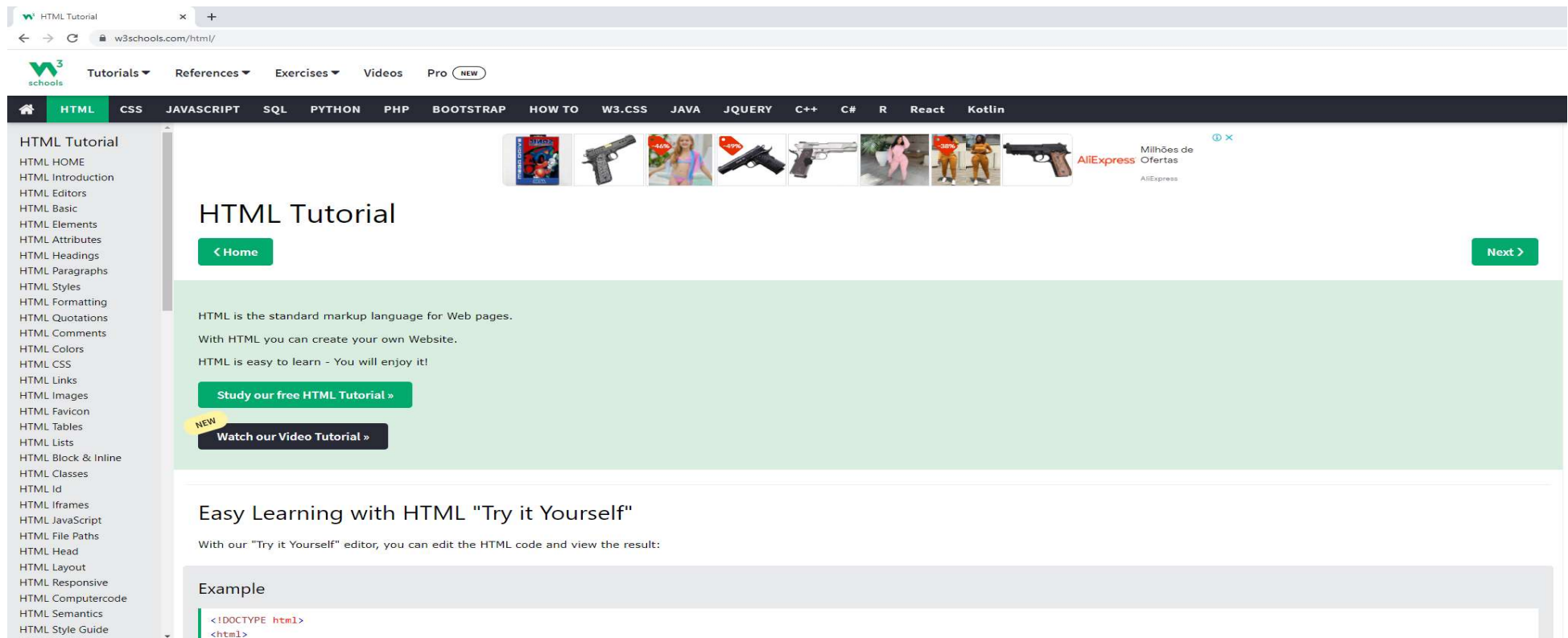
tag de abertura do link

tag de fechamento

- Sempre que possível o texto do link deve explicar para onde os visitantes serão levados
- Para escrever um bom texto de link, pense nas palavras que as pessoas podem utilizar ao pesquisar a página de destino do link

W3schools - html

<https://www.w3schools.com/html/>



TED (desafio simples)

Requisitos:

- Criar duas páginas
 - Index e “pagina2”;
 - Tema livre para cada página;
 - Deve haver um link redirecionando de uma página para a outra;
 - Escolher dois elementos novos no W3schools.
- Criar duas páginas
 - Index e “página 2”;
 - Utilizar os elementos vistos em sala:
 - Estrutura completa
 - Cabeçalho
 - Parágrafo e suas formatações básicas:
 - Negrito
 - Itálico
 - Quebra de linha
 - Sublinhado
 - Sobrescrito
 - Maior
 - Menor
 - Linha Horizontal
 - Estilos
 - Listas
 - Links

TED (desafio simples) - TQF

- Tempo:
 - Tempo de aula;
 - Entregar até próxima aula 31/03/2022.
- Qualidade:
 - Precisa funcionar;
 - As páginas precisam ter sentido com o tema (Trabalho criativo);
- Forma:
 - Anexar a atividade no Classroom (separada ou compactada);
 - Equipes de até 5 Pessoas.

Tabelas

- Podemos criar tabelas usando a tag `<table>`.
 - Cada linha deve ser declarada com a tag `<tr>` e cada célula com a tag `<td>`.
 - Opcionalmente podemos usar a tag `<th>` para declarar células que representem a linha de cabeçalho.

<code><table></code>			
<code><tr></code>	<code><th></th></code>	<code><th></th></code>	<code><th></th></code>
<code><tr></code>	<code><td></td></code>	<code><td></td></code>	<code><td></td></code>
<code><tr></code>	<code><td></td></code>	<code><td></td></code>	<code><td></td></code>
<code></table></code>			

Formulários

- Locais onde o usuário possa interagir com a página e enviar informações para serem processadas.
- Exemplo:
 - Formulário de Cadastro
 - Formulário de Login
 - Campo de Pesquisa

Formulário de Inscrição

file:///D:/Projetos%20WEB/Projeto%20Cu

Formulário de Inscrição

Preencha seus dados

Nome:

E-mail:

Endereço:

Cidade:

Telefone:

Usuário:

Senha:

Google Images

Favoritos

Formulário de Login

Formulário de Login

Informe seu e-mail e senha

E-mail:

Senha:

Google Images

Formulários

- Um formulário HTML é feito de um ou mais widgets.
- Esses widgets podem ser campos de texto (linha única ou de várias linhas), caixas de seleção, botões, checkboxes ou radio buttons.
- Na maior parte do tempo, estes elementos possuem uma legenda que descreve o seu objetivo.

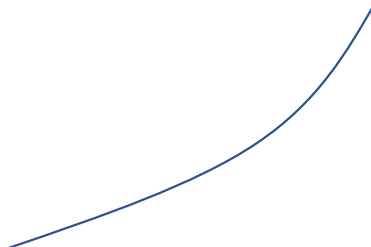
Formulários

- Tag `<form>`

`<form>`

`</form>`

Todos os campos do formulário devem ficar dentro da tag form.



- A tag `<form>` não apresenta uma referência visual quando criada, mas representa semanticamente um formulário.

Atributos

- Atributos da tag `<form>`
 - `autocomplete="on|off"`
 - A medida em que usamos os campos de um formulário o navegador irá começar a sugerir valores para o que estamos digitando, com base em valores que foram digitados anteriormente.
 - `action="string"`
 - Este atributo especifica para onde os dados do formulário serão enviados. Normalmente será o link de uma página no servidor:
 - `method="get|post"`
 - Especifica o método do HTTP que será utilizado para a submissão (get ou post):
 - `novalidate`
 - Impede a validação dos campos antes da submissão

Atributos

- Atributos da tag `<form>`
 - `target="_blank|self"`
 - Assim como na tag `<a>`, o `target` indica onde a página de destino será aberta.
 - `target="_blank"` serve para abrir um link em uma nova aba do navegador.

Elementos

- Elementos (Widgets) que podem ser utilizados no formulário:
 - Os formulários possuem campos onde inserimos dados. Os campos mais comuns dos formulários são criados pela tag `<input />`, que tem o atributo `type` onde indicamos o tipo de campo que queremos.

Elementos

- **Inputs de Texto**

- Para escrever um pequeno texto é utilizado a tag `<input/>` (auto close)

```
<form>  
  <input type="text" name="nome"/>  
</form>
```



Elementos

- **Atributos basicos tag** `<input />`
 - `type = "text"` (valor padrão é o text)
 - `placeholder`
 - texto que aparece quando o campo está vazio;
 - `maxlength`
 - quantidade máxima de caracteres
 - `required`
 - indica que o campo é necessário, não permitindo submeter o formulário se o campo estiver em branco;
 - `disabled`
 - deixa o campo desabilitado;
 - `readonly`
 - impede edições no campo, permitindo apenas que o usuário leia o valor.

Elementos

- **Email, Search e URL**

```
<input type="email" />  
<input type="search" />  
<input type="url" />
```

- Esses tipos são parecidos com o tipo “text”
- Teclados de tablets e smartphones se adaptam a cada tipo de input
- Tipos como **email** e **url** possuem validação

Elementos

- **Exemplos**

- Validação Email e URL

Email



Inclua um "@" no endereço de e-mail. "joao_20" está com um "@" faltando.

joao_20@gmail.com

Url

Enviar



Insira um URL.

http://www.submarino.com

Elementos

- **Number**

- Indica um campo do tipo number. Em computadores o campo fica com botões para aumentar e diminuir o número. Em dispositivos com teclado virtual é exibido um teclado numérico.

```
<form>  
  <input type="number" name="numero"/>  
</form>
```



Elemento

- **Radio Button**

- Cria um Radio Button, onde podemos selecionar apenas uma opção. Para obter a seleção exclusiva é necessário colocar o mesmo atributo “name”. Isso faz com que o navegador entenda que ambos pertencem ao mesmo campo. Se passarmos nomes diferentes poderíamos selecionar os dois valores, o que não é o objetivo dos Radio Buttons.

```
1 <form>
2   <input type="radio" name="videogame"/> Atari <br>
3   <input type="radio" name="videogame"/> Super Nintendo <br>
4   <input type="radio" name="videogame"/> Playstation <br>
5 </form>
```

☐ Atari
☐ Super Nintendo
☐ Playstation

Elementos

- **Atributos basicos tag** `<input type="radio" />`
 - `checked="checked"`
 - Selecciona um elemento no checkbox

```
1 <form>
2   <input type="radio" name="videogame" checked/> Atari <br>
3   <input type="radio" name="videogame"/> Super Nintendo <br>
4   <input type="radio" name="videogame"/> Playstation <br>
5 </form>
```

- ☒ Atari
- ☐ Super Nintendo
- ☐ Playstation

Elemento

- **Checkbox**

- Diferente do Radio Button, onde só é possível selecionar uma opção, o CheckBox serve para selecionarmos mais de uma opção.

Também é possível utilizar o atributo checked

```
1 <form>
2   <input type="checkbox" name="atari" checked/> Atari <br>
3   <input type="checkbox" name="super_nintendo"/> Super Nintendo <br>
4   <input type="checkbox" name="playstation"/> Playstation <br>
5 </form>
```

☒ Atari
☐ Super Nintendo
☐ Playstation

Elemento

- **Date, Time e Datetime-local**
- **Datetime-local**
 - Serve para selecionar o tempo (horas, minutos).

```
<form>  
  <input type="datetime-local" name="datatempo"/>  
</form>
```

dd/mm/aaaa --:--

abril de 2019 ▼

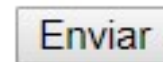
dom	seg	ter	qua	qui	sex	sáb
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

Elemento

- **Button, reset e submit**

- Esses tipos criam um botão.

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



- O tipo “submit” é usado para que o usuário clique quando quiser enviar o formulário.
- O tipo “reset” serve para quando o usuário quiser reiniciar o formulário. Cuidado, pois ao clicar o formulário inteiro será apagado!

Recebendo o conteúdo

- GET x POST
 - O método GET, transfere os dados para o serviço através da URL, e a quantidade de caracteres é limitado.
 - Já o método POST, você pode realizar envios mais seguros, e não há limite para transmissão de informações.
- No entanto, para processar dados de um formulário com o método POST, você precisa de algum tipo de servidor ou script de servidor para receber e processar os dados.
 - HTML por si só não é capaz de fazer isso.

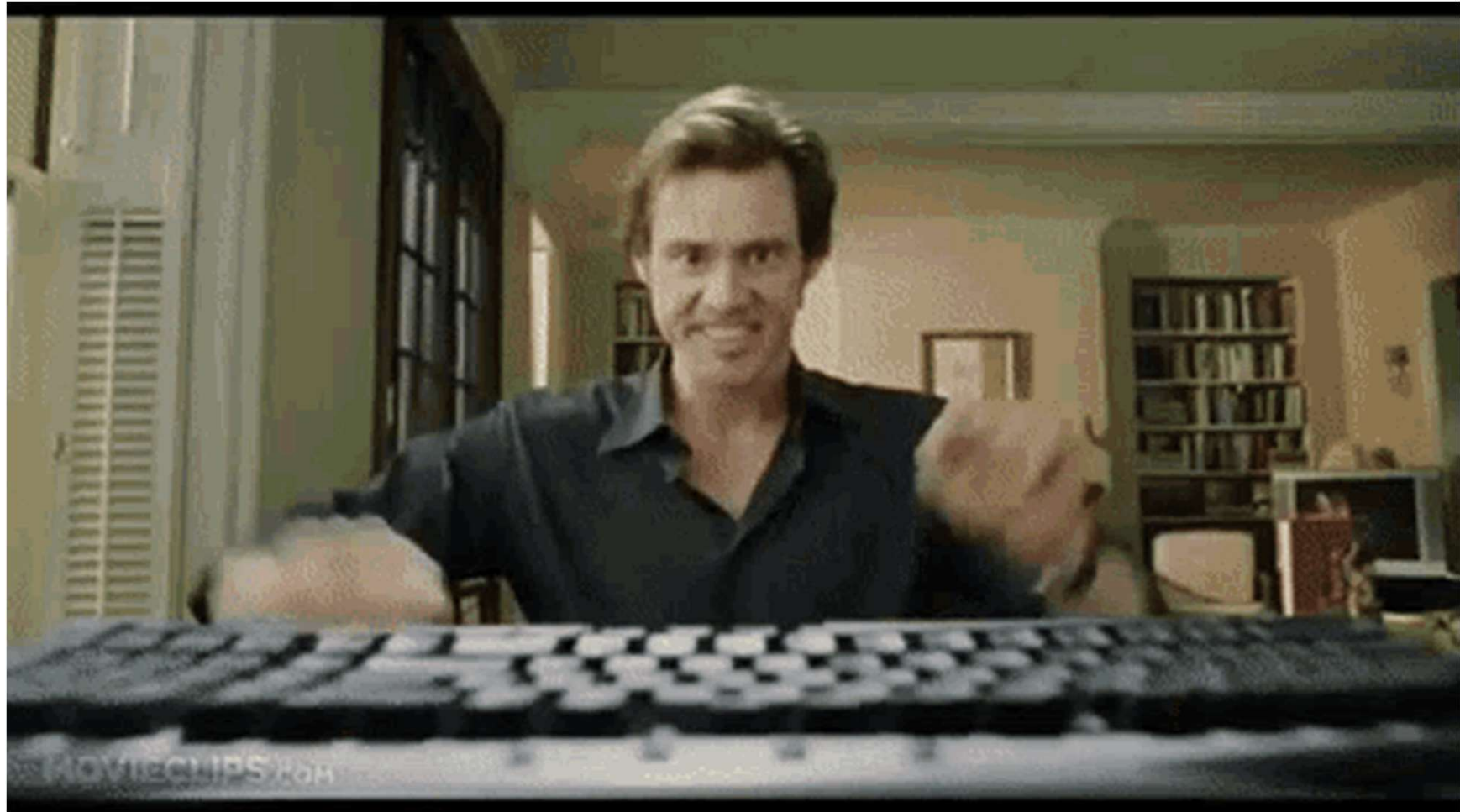
Recebendo o conteúdo

- HTML é uma **linguagem de marcação** que é usada para criar a estrutura de uma página da web, mas não possui a capacidade de processar formulários ou executar lógica do lado do servidor.
- O método POST é usado para enviar dados para o servidor, onde um script de servidor (por exemplo, PHP, Node.js, Python, Ruby) é necessário para processar esses dados.

Recebendo o conteúdo

- APIs da WEB
 - <https://developer.mozilla.org/pt-BR/docs/Web/API>
- Modelo de Objeto de Documento (DOM)
 - [https://developer.mozilla.org/pt-BR/docs/Web/API/Document Object Model](https://developer.mozilla.org/pt-BR/docs/Web/API/Document_Object_Model)

Tratamento do conteúdo

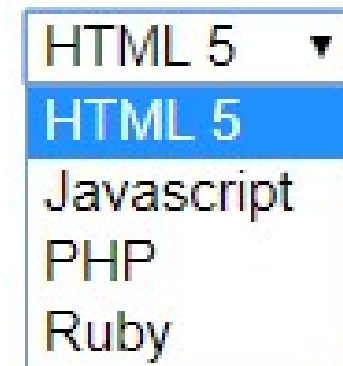


Elemento

- **Select**

- A tag <select> nos permite criar uma lista onde podemos selecionar um item. Cada uma das opções é criada pela tag <option>. O atributo “value” da tag option é o valor a ser enviado ao servidor, e o valor escrito dentro da tag é o que será exibido ao usuário.

```
<form>
  <select>
    <option value="html5">HTML 5</option>
    <option value="javascript">Javascript</option>
    <option value="php">PHP</option>
    <option value="ruby">Ruby</option>
  </select>
</form>
```



Elemento

- **Atributos basicos tag** <select>

- selected
 - Seleciona um elemento no select

```
<form>
  <select>
    <option value="html5">HTML 5</option>
    <option value="javascript" selected>Javascript</option>
    <option value="php">PHP</option>
    <option value="ruby">Ruby</option>
  </select>
</form>
```

Javascript ▼

Elemento

- **Fieldset e Legend**

- Serve para criarmos um campo dentro do formulário para poder melhor organizar os campos. A tag <legend> serve para darmos um título à seção.

Cadastro

Nome:

Email:

```
<form>
  <fieldset>
    <legend>Cadastro</legend>
    Nome: <input type="text"><br>
    Email: <input type="text"><br>
  </fieldset>
</form>
```


Elemento

- **Label**

- Esta tag serve para darmos nomes aos campos. Se passarmos o id do input no atributo “for”, estaremos indicando ao navegador que o Label é referente àquele campo. Quando o usuário clicar no texto do Label, o input receberá foco, permitindo que o usuário já entre com os dados.

```
<form autocomplete="on">  
  <label for="email">Email</label>  
  <input type="text" id="email">  
</form>
```

Email

Thanks!

DÚVIDAS?



@oanjoeducador

prof2132@iesp.edu.br

+55 81 98808 1008

angelofdiasg.tech (em dev)



<https://www.linkedin.com/in/angelofdiasg/>