



Aula 2

HTML & CSS

+spoilers



Vamos ver

- Vamos ver
 - HTML
 - Definição
 - Elementos básicos
 - Entrada de Dados
 - CSS
 - Necessidade
 - DOM
 - UX
 - Exercício
 - Javascript (Spoiler)
 - V8
 - Node



HTML

HTML

- Hyper Text Markup Language (HTML);
- Feita inicialmente para cientistas (**era simples**) [[CERN](#) de Genebra]
- Linguagem de Marcação (descritora);
- Descreve a estrutura de uma página através de elementos, ou tags;
- Originalmente feita para ser estática;

- Mãos à Obra
 1. Abrir um arquivo de texto em branco
 2. Salvar como index.html
 3. Abrir no VSCode



Primeira Página

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5  |   <title>Título</title>
6  </head>
7
8  <body>
9  |
10 |   <h1>Um Cabeçalho</h1>
11 |   <p>Primeiro Parágrafo</p>
12 |
13 </body>
14
15 </html>
```

`<!DOCTYPE html>` - Indica tipo do documento

`<html>` - tag de início de uma página

***Lembrar que tags devem ser fechadas**

`<head>` - Para alterar o nome da aba

`<title>` - Observe o nome da Aba

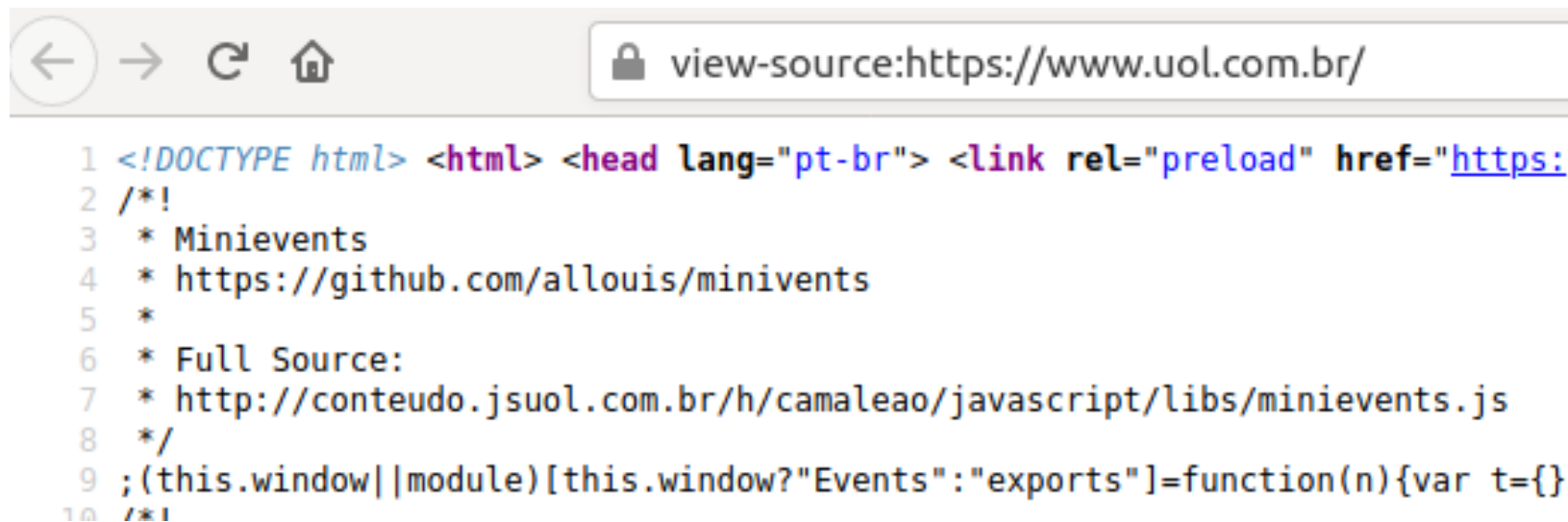
`<body>` - O que estará no corpo da página

`<h1>` - Cabeçalho

`<p>` - Parágrafo

[Validador HTML](#)

1. Vá em qualquer site;
2. Clique com o botão direito do Mouse;
3. Selecione a opção: Exibir código fonte
 1. Observe que **todas** as páginas têm a mesma estrutura



```
1 <!DOCTYPE html> <html> <head lang="pt-br"> <link rel="preload" href="https:
2 /*!
3 * Minievents
4 * https://github.com/allouis/minivents
5 *
6 * Full Source:
7 * http://conteudo.jsuol.com.br/h/camaleao/javascript/libs/minievents.js
8 */
9 ;(this.window||module)[this.window?"Events":"exports"]=function(n){var t={
10 /*!
```

- Agora é só uma questão de aprender os elementos e o colocar na ordem de interesse.
- [Bom ponto inicial](#)
- **Cabeçalhos <h*>**

```
<h1>Meu título principal</h1>
```

```
<h2>Meu título de alto nível</h2>
```

```
<h3>Meu subtítulo</h3>
```

```
<h4>Meu segundo subtítulo</h4>
```

Parágrafos e Listas

- Parágrafo

`<p>Este é um parágrafo simples</p>`

- Listas

- Ordenadas ` `

```
<ol>
  <li>banana</li>
  <li>maçã</li>
  <li>uva</li>
</ol>
```

- Não-ordenadas ``

```
<ul>
  <li>banana</li>
  <li>maçã</li>
  <li>uva</li>
</ul>
```


- Uma tabela é um conjunto estruturado de dados constituído por linhas e colunas;
- Usamos quando queremos apresentar conjuntos de dados correlatos de forma estruturada;
- Quando **não** usar tabelas:
 - Para formatar o layout de uma página;
- Dentro de uma tabela tem linhas e dentro das linhas temos células;

`<table></table>` = Marcação de tabela

`<tr>` = Linha (table row)

`<td>` = Célula (table data)

Tabelas: Propriedade

- **border** = propriedade de elemento **<table>**
 - Se igual a algum valor a tabela fica com bordas cada vez mais destacadas.
- **<th>** = Cabeçalho da tabela, aplicado dentro de elemento **<tr>**
- **colspan** = Tem a propriedade de mesclar células, propriedade de elemento tipo **<td>**

Cabeçalho 1	Cabeçalho 2	Cabeçalho 3	Cabeçalho 4
Primeira Célula			Quarta Célula
Quinta Célula	Sexta Célula	Sétima Célula	Oitava Célula

- **<i>** = Texto em Itálico (é usado para transmitir um significado tradicionalmente transmitido por itálico: palavras estrangeiras, designação taxonômica, termos técnicos, um pensamento...)
- **** = Texto em negrito (é usado para transmitir um significado tradicionalmente transmitido por negrito: palavras-chave, nomes de produtos, sentença principal...)
- **<u>** = sublinhado (é usado para transmitir um significado tradicionalmente transmitido pelo sublinhado: nome próprio, erro de ortografia...)
- **** = tem aparência de negrito mas além disso, leitores de sites falam em uma entonação diferente.

- Em um link queremos que ao se clicar em um elemento (um texto, uma imagem, ou outro elemento HTML) o Browser seja direcionado a outra URL;
- Logo, links são sempre aninhados a outros elementos HTML;
- Esta URL pode ser uma página ou um arquivo no servidor;
- Utiliza-se a tag ``

```
<p>  
  Estou criando um link para  
  <a href="https://www.cnn.com"  
  title="Uma fonte de informações!">a CNN USA</a>.
```

```
</p>  
<p>  
  Estou criando um link para  
  <a href="./em_branco.pdf"  
  title="Uma fonte de informações!">um arquivo no meu servidor</a>.
```

```
</p>
```

Exercício

- Vamos criar uma página HTML que seria o seu currículo;
- Ela deve conter:
 - Título identificando que é um currículo (cabeçalho h1);
 - Nome completo, idade (parágrafos)
- Endereço
 - Lista não-ordenada de itens contendo experiência.



CSS

- **Cascading Style Sheets (CSS)**
- Útil para separar da página o negócio e o estilo;
- Podemos controlar o layout da página todo em um só lugar e não espalhado pelo elemento.
- Boa fonte
- Poderíamos fazer:

```
<h1 style="color: red"> Um Cabeçalho </h1>
```

Ou criar um espaço para estilo dentro do head:

```
</head>
```

```
<style>  
  body {background-color: red;}  
  h1 {color: blue;}  
  p {color: green;}  
</style>
```

```
</head>
```

- O arquivo CSS ainda pode ser externo

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

```
<link rel="stylesheet" href="externo.css">
```

```
</head>
```

```
<body>
```

```
  <h1>Um Cabeçalho</h1>
```

```
</body>
```

```
</html>
```


- Podemos ainda definir uma classe de estilo no arquivo CSS

```
.minhaClasse {  
    color: red;  
}
```

- E no elemento que queremos aplicar a classe de estilo colocamos

```
<h1 class="minhaClasse">Um Cabeçalho</h1>
```

- O elemento formulário é utilizado para entrada de dados em combinação com outros elementos de entrada dentro dele.

```
<form>
```

```
.
```

```
form elements
```

```
.
```

```
</form>
```

- [Fonte boa](#)

Entrada de Datos

- `<input>`

- `<label>`

First name:

- `<select>`

Choose a car: Volvo ▾ Enviar

- `<textarea>`

The cat was playing in the garden.

- `<button>`

Enviar

```
<form id="loginForm">
  <label for="fname">Nome:</label><br>

  <input type="text" id="fname" name="fname" value="João"><br>

  <label for="lname">Sobrenome:</label><br>

  <input type="text" id="lname" name="lname" value="Silva"><br><br>

  <input type="submit" value="Enviar" onclick="alerta()">

</form>
```

```
<script>
function alerta() {
  loginForm = document.getElementById("loginForm");
  alert(loginForm.outerHTML)
}
</script>
```

Formulários HTML

Nome:

Sobrenome:

- `onclick=""`: Executa uma ação após o elemento ser clicado
- No exemplo anterior

```
<input type="submit" value="Enviar" onclick="alerta()">
```

- Criar então área de Script Javascript

```
<script>
```

```
function alerta(){  
    alert('Cliquei em Enviar')  
}
```

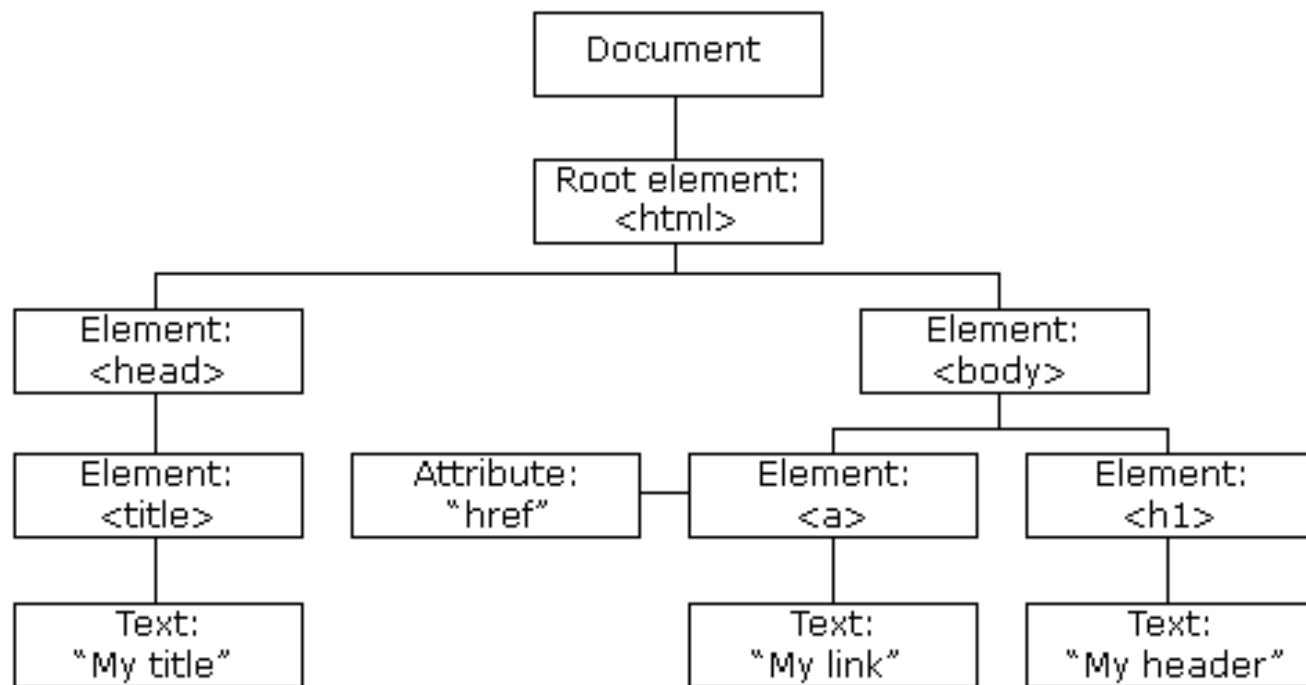
```
</script>
```

Ações (Dinâmicas + DOM)

- Agora estamos chegando em um outro ponto, onde a página **Estática** tem também comportamento **Dinâmico**.
- Esta parte dinâmica é feita utilizando JavaScript acessando o DOM.
- DOM: Document Object Model
 - Quando uma página é carregada, o Browser constrói uma representação interna hierárquica daquela página.
 - Aperte F12 em uma página e observe a primeira Aba carregada

Javascript será nossa próxima aula!!

DOM (por baixo dos panos)



- Javascript pode manipular o DOM
 - Criar elementos
 - Alterar Elementos
 - Deletar Elementos

```
<body>
  <h1>Evento onclick </h1>
  <p>Clique no botão para ver Javascript alterar o DOM e a
página</p>
  <button onclick="myFunction()">Click me</button>
  <p id="demo"></p>
  <script>
    function myFunction() {
      document.getElementById("demo").innerHTML = "Hello World";
    }
  </script>
</body>
```


- Observe que o JAVASCRIPT utiliza o id da tag para saber qual elemento vai recuperar;
 - Recuperar elemento por Id: `getElementById`
 - Recuperar elemento por nome da classe:
`getElementsByClassName`
 - Recuperar elemento por tipo da tag HTML:
`getElementsByTagName`
- Uma boa fonte sobre o DOM
- Manipulando o DOM de forma **dinâmica** podemos melhorar a qualidade do site melhorando a **experiência do usuário**;

UX = User Experience

- Chegamos a um ponto onde a WEB deixaria de ser usada não apenas por cientistas mas também pelo público geral;
- Os sistemas não poderiam continuar "pobres", "estáticos" e "poluídos";
 - Pop-ups;
 - Flash Player;
 - Applets...
- UX = o processo que os times de desenvolvimento tem para prover aos usuários experiências relevantes e significativas aos usuários.
 - Aquisição
 - Integração do produto
 - Branding
 - Design
 - Usabilidade
 - Funcionalidade



UX

- Muitas vezes o termo UX é utilizado para substituir os termos
 - UI Design
 - Usabilidade
- Contudo estes são apenas uma parte da UX;
"Nenhum produto é uma ilha. Um produto é maior que o produto" - Don Norman inventor do termo user experience
- Um produto é um conjunto coeso e integrado de experiências.
- Designers de UX não se concentram **apenas** em produtos usáveis.
- Focam em outros aspectos:
 - Prazer
 - Eficiência
 - Diversão



Apple AirPods
(3a geração)...

R\$ 1.899,00

Amazon.com...

Frete grátis



Javascript

- Javascript: A Lógica que permite atualizar o conteúdo de uma página **dinamicamente** (DOM)
 - Quem sabe JAVA, não irá estranhar.

HTML < Javascript < Node < Angular

- V8 (Engine Javascript usado no Chrome)
 - Escrito em C++;
 - Utilizado no Chrome (e clones);
 - MS passou a utilizar o V8 no Edge em 2020 (Chakra deprecated);
 - Firefox utiliza o SpiderMonkey.



Javascript + Node

- Antigamente era muito comum a separação clara entre Frontend e Backend
 - No Frontend era utilizado: HTML + CSS + Javascript
 - No Backend usava-se JAVA, Python, C++, C#...
- Assim os devs Front e Backend não se misturavam... até que...

Vamos arranjar uma forma de rodar Javascript **TAMBÉM** no servidor! Node!

- Arrancaram o V8 do Chrome e permitiram que ele rodasse FORA de um Browser.





Obrigado

Fernando Raposo
Analista - SERPRO

