

# Desenvolvimento Web

## Aula 3

Prof. Calebe Conceição

# O que vimos na última aula?

Aprendemos os primeiros passos para desenvolver nossas páginas web

Conhecemos os ambientes de desenvolvimento

Construímos nossas primeiras páginas com HTML

Entendemos o que é o DOM

Realizamos práticas simples e enviamos para o servidor, para acesso público

# O que vamos ver hoje:

Conheceremos alguns dos elementos principais das páginas web.

Listas

Imagens

Tabelas

Formulários

# Listas

# Lista

- Como vocês já viram em vários sites, existem listas
  - A ideia de lista é conseguir colocar elementos um abaixo do outro
  - Não ordenados
  - Ordenados

The screenshot displays the official website of the Universidade Federal de Sergipe. The header features the university's logo and name, along with navigation links for 'Página Inicial', 'WEBMAIL', and 'Mapa do Site'. A search bar is present with the placeholder text 'O que deseja buscar?' and a 'BUSCAR' button. Below the header, a horizontal menu lists various university departments: Institucional, Campi, Hospitais, Ensino, Pesquisa e Inovação, Extensão, Assistência Estudantil, Serviços, Publicações, Bibliotecas, and Acesso à informação. A secondary navigation bar includes social media icons (Facebook, Twitter, YouTube, RSS, Instagram) and buttons for 'Coronavírus', 'Comunidade interna', 'Ações afirmativas', and 'Acessibilidade'. The main content area is divided into three columns. The left column, titled 'A PROEX', lists links for 'Manual de Extensão / Legislação e Normas', 'Dados Estatísticos', 'Coordenação de Atividades de Extensão - CAEX', 'Coordenação da Central de Estágios UFS - CENEUFS', 'Avaliação de Extensão', 'Coordenação de Tecnologias Sociais e Ambientais - CTSA', 'Centro de Empreendedorismo da UFS - CEMP', 'Centro de Cultura e Arte - CULTART', and 'Mérito em Artes e Cultura'. The middle column, titled 'Pró-Reitoria de Extensão e Cultura', features a blue header with the university's logo and a section titled 'APRESENTAÇÃO' by 'Profa. Dra. Josefa de Lisboa Santos'. It describes the PROEX as the unit responsible for planning, coordinating, and supervising extension and cultural activities, listing four main functions: I. coordinating and supervising actions of member units; II. acting with public sectors and organs for partnership implementation; III. establishing partnerships with companies, institutions, and NGOs; and IV. elaborating the annual activity plan. The right column, titled 'Sistemas integrados', lists 'SIGAA - Acadêmico', 'SIPAC - Administrativo', and 'SIGRH - Recursos Humanos'. Below this is a 'Links rápidos' section with links to 'Consulta pública de processos', 'Sei', 'Protocolo GOV.BR', 'Acesso Remoto', and 'Procedimentos Organizacionais'.

```
<p> Lista de compras </p>
```

```
<ul>
```

```
</ul>
```

# Listas

- Vamos criar uma lista de compras
  - Testem aí

`<p> Lista de compras </p>`

`<ul>`

`</ul>`

ul

## ➤ unordered list

- é o marcador que cria uma lista não ordenada
- os itens da lista vão dentro dele
- nesse ponto não temos nada na tela

# Listas

- Testem aí
  - O que aconteceu?

```
<p> Lista de compras </p>
<ul>
  <li>Café</li>
  <li>Pão</li>
  <li>Maionese</li>
</ul>
```



li

## ➤ list item

- Item da lista
- é um item da lista
- vem com um símbolo que marca os itens
- O que tem dentro dele vai ficar no item

```
<p> Lista de compras </p>  
<ul>  
    <li>Café</li>  
    <li>Pão</li>  
    <li>Maionese</li>  
</ul>
```

li

➤ Troquem as palavras por imagens dentro das listas

➤ Como fazer isso?

```
<p> Lista de compras </p>  
<ul>  
  <li>Café</li>  
  <li>Pão</li>  
  <li>Maionese</li>  
</ul>
```

li

➤ Troquem as palavras por imagens dentro das listas

➤ Como fazer isso?

```
<p> Lista de compras </p>
<ul>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

li

- Troquem as palavras por imagens dentro das listas
  - Como fazer isso?
- **Agora, ao clicar na imagem abrir o endereço da wikipedia de cada uma das palavras**
  - Como fazer?

```
<p> Lista de compras </p>
<ul>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

li

- Troquem as palavras por imagens dentro das listas
  - Como fazer isso?
- Agora, ao clicar na imagem abrir o endereço da wikipedia de cada uma das palavras
  - Como fazer?

```
<p> Lista de compras </p>
```

```
<ul>
```

```
  <li><a
```

```
href="https://pt.wikipedia.org/
wiki/Caf%C3%A9"></a></li>
```

```
  <li><a
```

```
href="https://pt.wikipedia.org/
wiki/P%C3%A3o"></a></li>
```

```
  <li><a
```

```
href="https://pt.wikipedia.or
g/wiki/Maionese"></a></li>
```

```
</ul>
```

# Listas

- Testem aí
  - O que aconteceu?

```
<p> Lista de compras </p>
```

```
<ol>
```

```
<li>Café</li>
```

```
<li>Pão</li>
```

```
<li>Maionese</li>
```

```
</ol>
```

ol

➤ **ordered list**

- lista ordenada
- só trocar o ul por ol
  - unordered por ordered
- agora apareceram números no lugar dos símbolos

```
<p> Lista de compras </p>
```

```
<ol>
```

```
<li>Café</li>
```

```
<li>Pão</li>
```

```
<li>Maionese</li>
```

```
</ol>
```

# Listas

- Testem aí
  - O que aconteceu?

```
<p> Lista de compras </p>  
<ol type="I">  
  <li>Café</li>  
  <li>Pão</li>  
  <li>Maionese</li>  
  <li>Cachaça</li>  
</ol>
```



# type

## ➤ Atributo type

- tipo de lista ordenada
- só vale para lista ordenada
- alterou para algarismos romanos
  - não vai só colocando I
  - I, II, III, IV, ...

```
<p> Lista de compras </p>  
<ol type="I">  
  <li>Café</li>  
  <li>Pão</li>  
  <li>Maionese</li>  
  <li>Cachaça</li>  
</ol>
```

# type

## ➤ Atributo type

- variações
  - type="1"
  - type="a" ou type="A"
  - type="i" ou type="I"
- testem aí alguns tipos diferentes

```
<p> Lista de compras </p>  
<ol type="I">  
  <li>Café</li>  
  <li>Pão</li>  
  <li>Maionese</li>  
  <li>Cachaça</li>  
</ol>
```

# start

## ➤ Atributo start

- define em que número que começa a lista
- começa em 10
  - e assim vai

```
<p> Lista de compras </p>  
<ol type="I" start="10">  
  <li>Café</li>  
  <li>Pão</li>  
  <li>Maionese</li>  
  <li>Cachaça</li>  
</ol>
```

```
<p> Lista de compras </p>
```

```
<dl>
```

```
</dl>
```

# Lista

➤ Testem aí

➤ O que aconteceu?

<p> Lista de compras </p>

<dl>

</dl>

dl

## ➤ **description list**

- lista de descrições
- usada para detalhar um tópico logo em seguida
- outro tipo de lista
- não faz nada até agora

# Lista

- Testem aí
  - O que aconteceu?

```
<p> Lista de compras </p>
<dl>
  <dt>Café</dt>
  <dt>Pão</dt>
  <dt>Maionese</dt>
  <dt>Cachaça</dt>
</dl>
```

```
<p> Lista de compras </p>
```

```
<dl>
```

```
<dt>Café</dt>
```

```
<dt>Pão</dt>
```

```
<dt>Maionese</dt>
```

```
<dt>Cachaça</dt>
```

```
</dl>
```

## ➤ definition term

- termo a definir
- coloca o termo a ser definido/detalhado
- da forma que tá
  - tá igual a ul

# Lista

- Testem aí
  - O que aconteceu?

```
<p> Lista de compras </p>
```

```
<dl>
```

```
  <dt>Café</dt>
```

```
    <dd>Quente e com  
leite.</dd>
```

```
  <dt>Pão</dt>
```

```
    <dd>Fatiado integral</dd>
```

```
  <dt>Maionese</dt>
```

```
    <dd>Vegana, sem gluten,  
sem gosto, sem nada</dd>
```

```
  <dt>Cachaça</dt>
```

```
    <dd>Pura!</dd>
```

```
</dl>
```



dd

➤ **definition description**

- descrição da definição
- termo que fica formatado para dentro do dt
- explica o dt

```
<p> Lista de compras </p>
```

```
<dl>
```

```
  <dt>Café</dt>
```

```
    <dd>Quente e com  
leite.</dd>
```

```
  <dt>Pão</dt>
```

```
    <dd>Fatiado integral</dd>
```

```
  <dt>Maionese</dt>
```

```
    <dd>Vegana, sem gluten,  
sem gosto, sem nada</dd>
```

```
  <dt>Cachaça</dt>
```

```
    <dd>Pura!</dd>
```

```
</dl>
```

# Imagens

# Colocar imagens na página

- **Todas as páginas que acessamos tem imagens**
  - Então temos que colocar imagens nas nossas páginas
  - Criem uma pasta **imgs** dentro do projeto de página de vocês
    - Nessa pasta vão ir todas as imagens que vamos usar
    - Cuidar para salvar as imagens nesta pasta

# Colocar imagens na página

- **Todas as páginas que acessamos tem imagens**
  - Acessem o Google
    - Cliquem em imagens ⇒ Ferramentas ⇒ tamanho ⇒ médio
    - Procurem por alguma imagem que vocês queiram
    - Salvem essa imagem na pasta **imgs** (segundo botão do mouse)
      - Com o nome **imagem1**

# img ⇒ imagem

## ➤ **Marcador para inserir uma imagem**

- inserir o código abaixo em um nova página html
  - dentro do <body>
- ****
  - verifiquem se a imagem aparece corretamente no navegador

# img ⇒ imagem

➤ ``

➤ um marcador que não precisa ser fechado

➤ `<img >`

➤ um atributo que define onde está a imagem

➤ **src ⇒ source**

➤ vai o caminho de onde está a imagem a ser exibida

# img ⇒ imagem

➤ ``

➤ alterando o conteúdo do **src**

➤ **altera a imagem que exibimos**

➤ **Peguem outra imagem**

➤ **salvem como imagen2.jpg**

➤ **Coloquem na pasta imgs**

➤ **troquem o conteúdo do src para mostrar a imagen2.jpg**

# img ⇒ imagem

## ➤ Testem no navegador

- ``
- um atributo novo
  - **alt** ⇒ alternative
    - alternativo
    - define o texto que aparece caso de problema na figura



# img ⇒ imagem

## ➤ **Testem no navegador**

- ``
- um atributo novo
  - O que aconteceu?

# img ⇒ imagem

- **alt** ⇒ alternative

- é importante

- mesmo quando a página carrega certo...

- Porque?

# img ⇒ imagem

## ➤ alt ⇒ alternative

- é importante

- mesmo quando a página carrega certo...

- Porque?

- **principalmente para pessoa que não enxerga**

- usam leitores de tela

- o texto do alt é lido para ele explicando o que tem na tela

# img ⇒ imagem

## ➤ **Atualizar o código da imagem para**

- ``
- O que mudou?

# img ⇒ imagem

## ➤ Atualizar o código da imagem para

- ``
- O que mudou?
  - A imagem ficou com uma largura específica

# img ⇒ imagem

## ➤ **Atributo width ⇒ largura**

- é o número de pixels que a imagem vai ocupar de largura
  - pixel é a unidade de medida da tela
    - possivelmente a tela de vocês tem 1080 pixels de largura
- se colocamos **width="100"**
  - quer dizer que a largura da imagem será de 100 pixels

# img ⇒ imagem

## ➤ **Atributo width ⇒ largura**

- o html encolhe a figura proporcionalmente a altura também
- se temos uma figura de 400 pixels de largura e 500 altura
  - colocamos width="100"
    - a altura passa para 125
    - então a imagem passa a ter
      - 100 pixels de largura e 125 altura automaticamente

# img ⇒ imagem

## ➤ **Atualizar o código da imagem para**

- ``
- O que mudou?



# img ⇒ imagem

## ➤ **Atualizar o código da imagem para**

- ``
- O que mudou?
  - A imagem ficou com uma altura específica

# img ⇒ imagem

## ➤ **Atributo height ⇒ altura**

- é o número de pixels que a imagem vai ocupar de altura
  - pixel é a unidade de medida da tela
    - possivelmente a tela de vocês tem 1080 pixels de largura
- se colocamos **height="100"**
  - quer dizer que a altura da imagem será de 100 pixels

# img ⇒ imagem

## ➤ **Atributo height ⇒ altura**

- o html encolhe a figura proporcionalmente a largura também
- se temos uma figura de 400 pixels de largura e 500 altura
  - colocamos height="100"
    - a largura passa para 80
    - então a imagem passa a ter
      - 80 pixels de largura e 100 altura automaticamente

# img ⇒ imagem

## ➤ **Atualizar o código da imagem para**

- ``
- O que aconteceu?

# img ⇒ imagem

## ➤ Atualizar o código da imagem para

- ``
- O que aconteceu?
  - a imagem ficou com 50 de altura e 50 de largura

# img $\Rightarrow$ imagem

- **Se colocarmos height e width juntos**

- o tamanho fica fixo
- a imagem pode ficar distorcida
  - se colocarmos tamanho “errados”
  - para maior também vale

# img ⇒ imagem

## ➤ Testem no navegador

➤ ``

# Colocar imagens na página

- Como fazer para colocar uma imagem do lado da outra?



# Colocar imagens na página

## ➤ Como fazer para colocar uma imagem do lado da outra?

➤ Colocar um marcador seguido do outro

➤ ``  
``

# Colocar imagens na página

- **Como fazer para colocar uma imagem embaixo da outra?**

# Colocar imagens na página

## ➤ Como fazer para colocar uma imagem embaixo da outra?

➤ Colocar um `<br>` entre as imagens

➤ ``  
`<br>`  
``

# Colocar imagens na página

- Como fazer para fazer com que ao clicar em uma imagem ir para outra página?

# Colocar links nas imagens

- **Como fazer para colocar uma imagem embaixo da outra?**
  - colocar um o `<img>` entre `<a>` `</a>` entre as imagens
  - `<a href="http://www.ifsul.edu.br">`  
``  
`</a>`

# Tabelas

# Tabelas HTML

- Bastante frequentes para organizar o conteúdo de uma página
- Obviamente tem linhas e colunas
- E tem células que são os conteúdos do par linha e coluna

globo.com

g1

globoesporte

gshow

videos

ASSINE JÁ

MINHA CONTA

E-MAIL

ENTRAR

MENU

ge

LIBERTADORES

🔍

BUSCAR

Simule

Simule os resultados e veja como seu time pode se classificar

FASE DE GRUPOS

GRUPO A

CLASSIFICAÇÃO	P	J	V	E	D	GP	GC	SG	%	ULT. JOGOS	
1 Internacional	0.0	13	5	4	1	0	9	4	5	86.7	●●●●●
2 River Plate	0.0	9	5	2	3	0	8	3	5	60	●●●●●
3 Palestino	0.0	4	5	1	1	3	5	6	-1	26.7	●●●●●
4 Alianza Lima	0.0	1	5	0	1	4	1	10	-9	6.7	●●●●●

6ª RODADA

TER 07/05/2019 MONUMENTAL DE NÓREZ 21:30

RIV vs INT

TER 07/05/2019 NACIONAL DE LIMA 21:30

ALI vs PLT

GRUPO B

CLASSIFICAÇÃO	P	J	V	E	D	GP	GC	SG	%	ULT. JOGOS	
1 Cruzeiro	0.0	15	5	5	0	0	10	0	10	100	●●●●●
2 Emelec	0.0	6	5	1	3	1	4	4	0	40	●●●●●
3 Deportivo Lara	0.0	5	5	1	2	2	4	7	-3	33.3	●●●●●
4 Huracán	0.0	1	5	0	1	4	2	9	-7	6.7	●●●●●

6ª RODADA

QUIA 08/05/2019 EL PALACIO 19:15

HUR vs LAR

QUIA 08/05/2019 MINERAO 19:15

CRU vs EME

# Tabelas HTML

- Tabelas devem ser usadas para casos em **que existem dados a serem tabulados!**
  - Não é para usar loucamente
  - em 2000-2005 era um problema
    - os desenvolvedores não tinham

globo.com

g1

globoesporte

gshow

videos

ASSINE JÁ

MENHA CONTA

E-MAIL

ENTRAR

≡

MENU

ge

LIBERTADORES

🔍

BUSCAR

Simule os resultados e veja como seu time pode se classificar

Simular

⏪

FASE DE GRUPOS

⏩

GRUPO A

CLASSIFICAÇÃO	P	J	V	E	D	GP	GC	SG	%	ULT. JOGOS	
1 Internacional	0.0	13	5	4	1	0	9	4	5	86.7	●●●●●
2 River Plate	0.0	9	5	2	3	0	8	3	5	60	●●●●●
3 Palestino	0.0	4	5	1	1	3	5	6	-1	26.7	●●●●●
4 Alianza Lima	0.0	1	5	0	1	4	1	10	-9	6.7	●●●●●

6ª RODADA

TER 07/05/2019 MONUMENTAL DE NÓREZ 21:30

RIV vs INT

TER 07/05/2019 NACIONAL DE LIMA 21:30

ALI vs RIV

QUI 08/05/2019 EL PALACIO 19:15

HUR vs LAR

QUI 08/05/2019 MINERAO 19:15

CRU vs EME



```
<table>  
</table>
```

# Tabelas HTML

- Testem aí
  - O que acontece?

```
<table>  
</table>
```

# Tabelas HTML

- Testem aí
  - O que acontece?
  - Não aparece nada
  - Marcadores
    - Abrir e fechar

# Tabelas HTML

```
<table>  
  <tr>  
  </tr>  
</table>
```

- Testem aí
  - O que acontece?
  - Ainda nada
  - **tr ⇒ table row**
    - linha de tabela
    - criou uma linha
      - mas não tem nada dentro

# Tabelas HTML

- Testem aí
  - O que acontece?

```
<table>  
  <tr>  
    <td> Linha1,Coluna1 </td>  
    <td> Linha1,Coluna2?</td>  
  </tr>  
</table>
```

# Tabelas HTML

- Testem aí
  - O que acontece?
  - **td** ⇒ **table data cell**
    - célula de dados da tabela
    - é um elemento da tabela
    - um par linha, coluna
    - fica dentro do **tr**

```
<table>
  <tr>
    <td> Linha1,Coluna1 </td>
    <td> Linha1,Coluna2?</td>
  </tr>
</table>
```

# Tabelas HTML

- Como que eu faço para adicionar mais uma linha?

```
<table>
  <tr>
    <td> Linha1, Coluna1 </td>
    <td> Linha1, Coluna2?</td>
  </tr>
</table>
```

# Tabelas HTML

- Como que eu faço para adicionar mais uma linha?
  - Adicionar um tr dentro da tabela
    - Cria mais uma linha
    - e se eu quiser mais uma linha?

```
<table>
  <tr>
    <td> Linha1, Coluna1 </td>
    <td> Linha1, Coluna2?</td>
  </tr>
  <tr>
    <td> Linha2, Coluna1 </td>
    <td> Linha2, Coluna2</td>
  </tr>
</table>
```

# Tabelas HTML

- Como que eu faço para adicionar mais uma linha?
  - Adicionar um tr dentro da tabela
    - Cria mais uma linha
    - e se eu quiser mais uma linha?
      - e assim vai...

```
<table>
  <tr>
    <td> Linha1,Coluna1 </td>
    <td> Linha1,Coluna2?</td>
  </tr>
  <tr>
    <td> Linha1,Coluna1 </td>
    <td> Linha1,Coluna2?</td>
  </tr>
  <tr>
    <td> Linha2,Coluna1 </td>
    <td> Linha2,Coluna2</td>
  </tr>
</table>
```



# Tabelas HTML

- O que acontece se eu fizer essa tabela?

```
<table>
  <tr>
    <td> Linha1, Coluna1 </td>
    <td> Linha1, Coluna2</td>
    <td> Linha1, Coluna3</td>
  </tr>
  <tr>
    <td> Linha2, Coluna1 </td>
  </tr>
  <tr>
    <td> Linha3, Coluna1 </td>
    <td> Linha3, Coluna2</td>
  </tr>
</table>
```

# Tabelas HTML

- O que acontece se eu fizer essa tabela?
  - Ficam os espaços vazios
  - Sempre vale o maior número de tds dentro das linhas de uma tabela
    - no caso 3 da primeira linha

```
<table>
  <tr>
    <td> Linha1, Coluna1 </td>
    <td> Linha1, Coluna2</td>
    <td> Linha1, Coluna3</td>
  </tr>
  <tr>
    <td> Linha2, Coluna1 </td>
  </tr>
  <tr>
    <td> Linha3, Coluna1 </td>
    <td> Linha3, Coluna2</td>
  </tr>
</table>
```

# Tabelas HTML

- O que acontece se eu fizer essa tabela?

```
<table>
  <tr>
    <td> Linha1, Coluna1 </td>
    <td> </td>
    <td> Linha1, Coluna3</td>
  </tr>
  <tr>
    <td> Linha2, Coluna1 </td>
  </tr>
  <tr>
    <td> </td>
    <td> Linha3, Coluna2</td>
    <td> Linha3, Coluna3 </td>
  </tr>
</table>
```

# Tabelas HTML

- O que acontece se eu fizer essa tabela?
  - Ficam espaços vazios também
  - Só que no meio da tabela
  - Os tds vazios ficam vazios
  - Tem de colocar eles para caso queiramos uma alinhamento dos seguintes...

```
<table>
  <tr>
    <td> Linha1, Coluna1 </td>
    <td> </td>
    <td> Linha1, Coluna3</td>
  </tr>
  <tr>
    <td> Linha2, Coluna1 </td>
  </tr>
  <tr>
    <td> </td>
    <td> Linha3, Coluna2</td>
    <td> Linha3, Coluna3 </td>
  </tr>
</table>
```

# Tabelas HTML

- Testem esse outro tipo de linha
  - O que aconteceu aí?

```
<table>
  <tr>
    <th> Coluna1 </th>
    <th> Coluna2 </th>
  </tr>
  <tr>
    <td> Linha2,Coluna1 </td>
    <td> teste </td>
  </tr>
  <tr>
    <td> Linha3,Coluna2</td>
    <td> Linha3,Coluna3 </td>
  </tr>
</table>
```

# Tabelas HTML

- Testem esse outro tipo de linha
  - O que aconteceu aí?
  - **th** ⇒ **table header**
    - cabeçalho de tabela
    - tem uma formatação diferente
    - centralizado e negrito

```
<table>
  <tr>
    <th> Coluna1 </th>
    <th> Coluna2 </th>
  </tr>
  <tr>
    <td> Linha2,Coluna1 </td>
    <td> teste </td>
  </tr>
  <tr>
    <td> Linha3,Coluna2</td>
    <td> Linha3,Coluna3 </td>
  </tr>
</table>
```

# Tabelas HTML

- Testem esse outro tipo de linha
  - O que aconteceu aí?

```
<table>  
    <caption> Tabela de teste  
</caption>  
    <tr>  
        <th> Coluna1 </td>  
        <th> Coluna2 </td>  
    </tr>  
    <tr>  
        <td> Linha2,Coluna1 </td>  
        <td> teste </td>  
    </tr>  
    <tr>  
        <td> Linha3,Coluna2</td>  
        <td> Linha3,Coluna3 </td>  
    </tr>  
</table>
```

# Tabelas HTML

- Testem esse outro tipo de linha
  - O que aconteceu aí?
  - **caption ⇒ legenda**
    - Coloca uma legenda como em tabelas normais
    - Tem de aparecer logo depois do `<table>`

```
<table>
  <caption> Tabela de teste
</caption>
  <tr>
    <th> Coluna1 </td>
    <th> Coluna2 </td>
  </tr>
  <tr>
    <td> Linha2,Coluna1 </td>
    <td> teste </td>
  </tr>
  <tr>
    <td> Linha3,Coluna2</td>
    <td> Linha3,Coluna3 </td>
  </tr>
</table>
```



# Tabelas

- Atributos de tabelas
  - Vão ver tabelas com width, height, color, ...
  - Tem vários atributos
    - (quase) Todos passaram a não ser suportados pelo HTML5
    - Ou seja, não é recomendável mais fazer no html
      - deixar essas coisas para os estilos do CSSs

Criar uma tabela

Visualização de banco de dados

# Tabela de usuários

## ➤ Uma tabela

### ➤ Colunas

- Identificador
- Nome
- Sobrenome
- Email
- Data de nascimento

## ➤ Colocar 4 usuários na tabela

- Preencher com dados “reais”
- Por exemplo

Id	Nome	Sobrenome	Email	Nascimento
1	Arnaldo	Silva	arnaldinho83@bol.com.br	27/10/1936
2	Carmen	Aragão	caminha79@uol.com.br	02/03/1940
3	Leonidas	de Orleans	reideorleans@aol.com	05/07/1951
4	Clotilde	Batista	clobat@terra.com.br	11/12/1947

# Tabelas

- atributo **rowspan**
  - row ⇒ linha
  - aumentar a quantidade de células ocupadas
  - ocupar células para baixo

```
<table border="1">
  <caption> Tabela de teste
</caption>
  <tr>
    <th rowspan="2">Coluna 1 e 2
  </th>
    <th> Coluna2 </th>
  </tr>
  <tr>
    <td> Linha2,Coluna1 </td>
    <td rowspan="3"> teste </td>
  </tr>
  <tr>
    <td> Linha3,Coluna2</td>
    <td> Linha3,Coluna3 </td>
  </tr>
</table>
```

# Tabelas

- Testem esse outro tipo de linha
  - O que aconteceu aí?

```
<table border="1">
  <caption> Tabela de teste
</caption>
  <tr>
    <th colspan="2">Coluna 1 e 2
  </th>
    <th> Coluna2 </th>
  </tr>
  <tr>
    <td> Linha2,Coluna1 </td>
    <td colspan="3"> teste </td>
  </tr>
  <tr>
    <td> Linha3,Coluna2 </td>
    <td> Linha3,Coluna3 </td>
  </tr>
</table>
```

# Tabelas

## ➤ atributo **colspan**

- **col** ⇒ **coluna**
- aumentar a quantidade de células ocupadas
- ocupar células para direita

```
<table border="1">  
  <caption> Tabela de teste  
</caption>  
  <tr>  
    <th colspan="2">Coluna 1 e 2  
</th>  
    <th> Coluna2 </th>  
</tr>  
<tr>  
  <td> Linha2,Coluna1 </td>  
  <td colspan="3"> teste </td>  
</tr>  
<tr>  
  <td> Linha3,Coluna2 </td>  
  <td> Linha3,Coluna3 </td>  
</tr>  
</table>
```

# Tabelas

- Atributos de tabelas
  - Vão ver tabelas com width, height, color, ...
  - Tem vários atributos
    - (quase) Todos passaram a não ser suportados pelo HTML5
    - Ou seja, não é recomendável mais fazer no html
      - deixar essas coisas para os estilos do CSSs

# Tabelas

## ➤ atributo **colspan** + **rowspan**

- Faz exatamente o que deve fazer
- aumentar a quantidade de células ocupadas para direita e para baixo

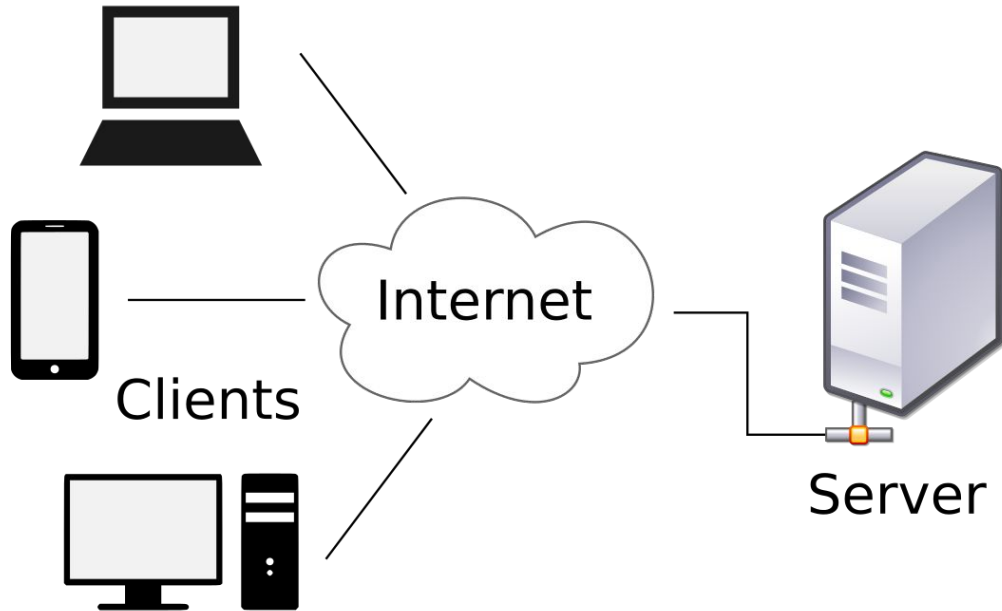
```
<table border="1">
  <caption>Tabela de teste
</caption>
  <tr>
    <th rowspan="2" colspan="2">
Coluna 1 e 2 </th>
    <th> Coluna2 </th>
  </tr>
  <tr>
    <td> Linha2, Coluna1 </td>
    <td rowspan="3" colspan="3">
teste </td>
  </tr>
  <tr>
    <td> Linha3, Coluna2 </td>
    <td> Linha3, Coluna3 </td>
  </tr>
</table>
```



# Formulários

- Lá na primeira aula...
  - Falamos que as páginas Web funcionam na arquitetura **cliente-servidor**
  - Os navegadores recebem informação de um servidor que processa os dados para serem renderizados na tela do navegador

# Formulários



# Formulários

- Praticamente todas as páginas Web precisam de alguma maneira que o usuário coloque dados para dentro do sistema
  - Enviando para o servidor
  - Para isso usamos **formulários**

The image shows a screenshot of a web application interface. At the top, there is a dark blue header bar with navigation links: "Página Inicial", "WEBMAIL", and "Mapa do Site". Below the header is a search bar with the placeholder text "O que deseja buscar?" and a red "BUSCAR" button. Underneath the search bar is a horizontal menu with links: "Serviços", "Publicações", "Bibliotecas", and "Acesso à informação". The main content area is a light gray box containing a white login form. The form features the "sei!" logo at the top. Below the logo are two input fields: one for the username "calebemicael" and another for the password "Senha". A blue "ACESSAR" button is positioned below the password field. At the bottom right of the login form, there is a link that says "Autenticação em dois fatores".

# form $\Rightarrow$ formulário

- O marcador **form**
  - é o marcador que cria um formulário
    - criem uma pasta nova (aula3)
    - criem uma página nova (formulario.html)
    - coloquem um formulário dentro
      - **<form> </form>**
      - **O que apareceu na página?**

# form $\Rightarrow$ formulário

- O marcador **form**
  - em si não faz nada
  - ainda não tem nada dentro do formulário
  - **é um agrupador de marcadores que serão mandados para o servidor**
    - Precisa ser aberto e fechado
    - Conteúdo dele é que constitui o formulário

# form $\Rightarrow$ formulário

- O marcador **form**

- Alguns marcadores específicos vão ter suas informações enviadas para o servidor
  - Podemos colocar todos os tipos de marcadores dentro do marcador form, mas **somente alguns** vão ser mandados para o servidor

# form $\Rightarrow$ formulário

- Testem um formulário
  - O que apareceu na tela?

```
<h1> Meu primeiro formulário</h1>
<form>
  Login:<input type="text"><br>
  Senha:<input type="text">
</form>
```

# form $\Rightarrow$ formulário

- Testem um formulário
  - O que apareceu na tela?

```
<h1> Meu primeiro formulário</h1>
<form>
  Login:<input type="text"><br>
  Senha:<input type="text">
</form>
```



# Input ⇒ Entrada de dados

- Um dos marcadores que define que determinado conteúdo vai ser enviado ao servidor
  - Não precisa ser fechado
  - Geralmente é acompanhado de um texto antes para informar para o usuário o que deve ser inserido no campo
    - `<input type="text">`

# Input $\Rightarrow$ Entrada de dados

- Atributo **type**

- Como vocês devem imaginar

- Informa o tipo que é o **input**

- Exemplo foi do tipo **text**

- Usado para inserir qualquer tipo de texto

# Input $\Rightarrow$ Entrada de dados

- Atributo **type**

- Como vocês devem imaginar
  - Variando o conteúdo do type
    - vai alterar o que aparece como resultado do input
    - existem variações do valor do type

# form ⇒ formulário

- O que está estranho nesse formulário?
  - Quando vocês vão fazer login em um site é assim que aparecem os campos?

```
<h1> Meu primeiro formulário</h1>
<form>
  Login:<input type="text"><br>
  Senha:<input type="text">
</form>
```

# Input ⇒ Entrada de dados

- Atributo **type = "password"**
  - Alterem a página para ver o que acontece

```
<h1> Meu primeiro formulário</h1>
<form>
  Login:<input type="text"><br>
  Senha:<input type="password">
</form>
```

# Input ⇒ Entrada de dados

- Atributo **type = “password”** ⇒ **senha**
  - Funciona exatamente igual ao text
  - Só que
    - Oculta os caracteres na hora que escrevemos
    - Como esperado para um campo de senha
  - Não quer dizer que é uma senha
    - Só é do tipo password

# Input $\Rightarrow$ Entrada de dados

- Vamos enviar para o servidor os atributos, certo?
  - Como sabemos qual valor é de qual atributo?

```
<h1> Meu primeiro formulário</h1>
<form>
  Login:<input type="text"><br>
  Senha:<input type="password">
</form>
```

# Identificar inputs

- Atributo **name**

- Testem no navegador de vocês...

```
<h1> Meu primeiro formulário</h1>
```

```
<form>
```

```
  Login:<input name="login" type="text"><br>
```

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

```
</form>
```



# Identificar inputs

- Não mudou nada visualmente
- Só mudou que os inputs agora tem um valor de nome
  - Isso ajuda para mandar a informação e para usar os próximos tipos de inputs

```
<h1> Meu primeiro formulário</h1>
```

```
<form>
```

```
  Login:<input name="login" type="text"><br>
```

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

```
</form>
```

# Identificar inputs

- Guardem isso na cabeça que veremos a utilidade deste **atributo name** em breve por outras razões

```
<h1> Meu primeiro formulário</h1>
```

```
<form>
```

```
  Login:<input name="login" type="text"><br>
```

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

```
</form>
```

# Input $\Rightarrow$ Entrada de dados

- Vamos enviar para o servidor os atributos, certo?
  - Como enviamos de fato para o servidor?
  - O que está faltando para enviar?

```
<h1> Meu primeiro formulário</h1>
```

```
<form>
```

```
  Login:<input  name="login"  type="text"><br>
```

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

```
</form>
```

# Input ⇒ Entrada de dados

- Testem no navegador de vocês

```
<h1> Meu primeiro formulário</h1>
```

```
<form>
```

```
  Login:<input name="login" type="text"><br>
```

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

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

```
</form>
```

# Input $\Rightarrow$ Entrada de dados

## ➤ **type="submit"**

- Cria um botão padrão
- Com submit escrito
  - Padrão
  - Se alteramos o navegador, PODE ser que o que está escrito no botão também seja alterado
  - Testem em navegadores diferentes...

# Input $\Rightarrow$ Entrada de dados

## ➤ **type="submit"**

- Envia o formulário para algum lugar (em breve)
- **Cliquem no botão submit**
  - **Olhem a barra de url**
  - Os valores atribuídos ao atributo **name** dos inputs são mostrados na url com os textos digitados neles
    - E o **input password**?

# Input ⇒ Entrada de dados

- Testem no navegador de vocês
  - O que faz esse botão novo?

```
<h1> Meu primeiro formulário</h1>
```

```
<form>
```

```
  Login:<input name="login" type="text"><br>
```

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

```
  <input type="submit"> <input type="reset">
```

```
</form>
```

# Input $\Rightarrow$ Entrada de dados

## ➤ **type="reset"**

- Cria outro botão padrão
  - Com **reset** escrito
  - Se alterar o navegador PODE alterar o que está escrito no botão
  - Ao clicar no botão, o navegador voltar os valores dos inputs para os valores iniciais ("limpa")



# Input ⇒ Entrada de dados

- Testem no navegador de vocês

```
<form>
```

```
  Login:<input name="login" type="text"  
value="usuário"><br>
```

```
  Senha:<input name="senha" type="password"  
value="1234567"><br>
```

```
  <input type="submit"> <input  
type="reset">
```

```
</form>
```

# Input $\Rightarrow$ Atributo Value

## ➤ **value $\Rightarrow$ valor**

- Define um valor padrão para o input
- Usado para facilitar a vida do usuário
  - Auto-preencher
  - Ou dar sugestões de preenchimento
- Útil também no input reset
  - Voltar para esses valores

# Formulário

- **Existem mais vários tipos de input**
  - Vamos ver mais uns exemplos
  - Para que servem
  - Como usar
  - Crescer o formulário da aula passada

# Input $\Rightarrow$ Testem

```
<form>
  Nome:<input name="nome" type="text" value="nome"><br>
  Senha:<input name="senha" type="password" value="senha"><br>
Função:
  <input type="radio" name="funcao" value="aluno"> Aluno
  <input type="radio" name="funcao" value="professor"> Professor
  <input type="radio" name="funcao" value="outro"> Outro <br>
  <input type="submit"><input type="reset">
</form>
```

# Radio

- **type ⇒ radio**

- **type="radio"**

- escolher um entre muitos
    - dar opções fixas para o usuário
    - Como isso é feito no código html?
      - Como ele sabe quais opções são do radio?

# Input $\Rightarrow$ Testem

```
<form>
  Nome:<input name="nome" type="text" value="nome"><br>
  Senha:<input name="senha" type="password" value="senha"><br>
  Função:
  <input type="radio" name="funcao" value="aluno"> Aluno
  <input type="radio" name="funcao" value="professor"> Professor
  <input type="radio" name="funcao" value="outro"> Outro <br>
  <input type="submit"><input type="reset">
</form>
```

# Radio

## ➤ Agrupados pelo atributo name

- name="funcao" que identifica são do mesmo radio
- Testem colocar outro radio

Time:

```
<input type="radio" name="time" value="sergipe"> Sergipe  
<input type="radio" name="time" value="confianca"> Confiança  
<input type="radio" name="time" value="outro"> Outro  
<input type="radio" name="time" value="nenhum"> Nenhum
```

# Radio

- **Cliquem no submit**
  - Qual o valor do input time?
  - Qual o valor do input funcao?



# Radio

## ➤ **Cliquem no submit**

- Qual o valor do input time?
- Qual o valor do input funcao?
  - Olhem na url depois de apertar o submit
    - Foi o valor do atributo **value** que pode não ter relação com o texto que vem depois do radio

## ➤ Substituir o radio do time por checkbox

- Poder escolher mais de um time
- Testem

Time:

```
<input type="checkbox" name="time" value="ita"> Itabaiana  
<input type="checkbox" name="time" value="lag"> Lagartense  
<input type="checkbox" name="time" value="outro"> Outro  
<input type="checkbox" name="time" value="nenhum"> Nenhum
```

➤ **Cliquem no submit**

- **olhem a url para verificar os valores do time**

Time:

```
<input type="checkbox" name="time" value="ita"> Itabaiana  
<input type="checkbox" name="time" value="lag"> Lagartense  
<input type="checkbox" name="time" value="outro"> Outro  
<input type="checkbox" name="time" value="nenhum"> Nenhum
```

# Checkbox

## ➤ Agrupados pelo atributo name

- name="contato" que identifica são do mesmo checkbox
- Testem colocar outro checkbox

Contato:

```
<input type="checkbox" name="contato" value="sms"> SMS
```

```
<input type="checkbox" name="contato" value="whats"> Whats
```

```
<input type="checkbox" name="contato" value="ligacao"> ligacao
```

```
<input type="checkbox" name="contato" value="nenhum"> Nenhum
```

# Checkbox

- **Checkbox ⇒ Caixa de checagem**
  - poder selecionar vários
  - praticamente igual ao radio

# Button

- **Além dos botões padrão**

- Adicionem o input

- `<input type="button" value="Não clique!">`

# Button

- **type = button ⇒ botão**
  - o **value** é o valor que vai aparecer dentro do botão
    - **value="Não clique!"**
  - é como os botões padrão
    - mas que não tem ação pré definida

# Button

- **type = button ⇒ botão**
  - o **value** é o valor que vai aparecer dentro do botão
    - **value="Não clique!"**
  - Se colocarmos values dentro dos botões padrão
    - eles também se alteram para o valor que colocarmos
      - mas não mudam o comportamento de reset ou submit
      - no futuro poderemos associar ações javascript a esses botões



# Entrada de dados

- Testem no navegador de vocês
  - O que faz esse botões padrão com valor customizado?

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

```
<input type="reset" value="limpar">
```

# Button

- Além dos botões padrão
  - Não faz nada...ficou parado na mesma página
  - Adicionem o input
    - `<input type="button" value="Não clique!">`

# Button

- Além dos botões padrão
  - Adicionem o input
    - `<input type="button" value="Não clique!"  
onclick="alert('falei para não clicar!')">`

# Button

- **type = button ⇒ botão**

- onclick é um atributo que diz o que fazer quando o botão é clicado
  - é Javascript o que tem dentro
  - isso é para um futuro um pouco distante, mas já tá aí para vocês verem
- agora vocês já criaram um alert de JS

```
<form>
```

```
    Login:<input name="login"  
type="text"><br>
```

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

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

```
</form>
```

## ➤ Formulário simples

- com os inputs
- consegue já "funcionar"
- não fazia nada no enviar/submeter

```
<form>
```

```
    Login:<input name="login"  
type="text"><br>
```

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

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

```
</form>
```

➤ Testem aí

➤ Qual a diferença?

- **Define a forma que o formulário vai ser enviado**

- São métodos HTTP

- Os mais usados são **get e post**

- Existem mais métodos http

- Vamos escolher qual usar com base no que vamos enviar

➤ **get** é o padrão do form

➤ se não tiver um method

➤ Quais vocês acham que são as limitações do get?



➤ **get** é o padrão do form

- se não tiver um atributo method declarado, então será assumido que é o get.
- Quais vocês acham que são as limitações do get?
  - Dados visualizados na url
  - Limitação de tamanho porque vai no url
  - Imagem enviar um texto (tamanho máximo da url é 2048)
- Usado para pegar (get) dados de algum lugar

## ➤ valores dos inputs no get

➤ aula03/form2.html?login=gui&senha=123456

➤ ? ⇒ informa que daqui para frente são pares (nome=valores)

➤ & ⇒ separa os pares

➤ login=gui

➤ senha=123456

## ➤ **Define a forma que o formulário vai ser enviado**

- post deve ser explicitamente colocado
  - **method="post"**
  - Dados não são visualizados na url
    - São enviados no corpo da solicitação http
    - Indicado para dados sensíveis
  - Sem limite de dados
    - Envio de arquivos

```
<form method="get"  
action="teste.html">
```

```
    Login:<input name="login"  
type="text"><br>
```

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

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

- Indicar para onde formulário deve ir depois de enviar
  - testem aí e cliquem no enviar

- **action define para onde o formulário vai depois de clicar no enviar**
  - atualiza a página para a url indicada
    - independente de ser a mesma página
  - geralmente vai ser para uma página com extensão **.php**
    - que vai processar o conteúdo do formulário
    - e devolver uma outra página diferente

- **Entrem na url** <https://www.google.com.br>
  - adicionem no fim da url
  - adicionem no fim da url /search?q=celular&sort=6
  - adicionem no fim da url /search?q=celular&lr=lang\_pt&tbs=qdr:y
- **Qual é o valor do action? Qual é o valor do method?**

```
<form method="get"  
action="teste.html"  
target="_blank">
```

- **Como abrir a página?**
- **Testem aí**
  - **O que aconteceu?**

```
    Login:<input name="login"  
type="text"><br>
```

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

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

➤ **Define como o action deve ser executado**

- Por padrão **target = "\_self"**
  - abre a página do action na mesma aba
- **target = "\_blank"**
  - abre a página do action em uma nova aba
- Tem mais valor possíveis



➤ **Funciona também em links**

➤ Testem aí

```
<form method="get"  
action="teste.html"  
target="_blank">
```

```
    Login:<input name="login"  
type="text"><br>
```

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

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

```
</form>
```

```
<a href="http://globo.com"  
target="_blank"> Globo! </a>
```

```
<form method="get"
action="teste.html" target="_blank">
  <fieldset>
    <legend> Entrar no site
  </legend>
    Login:<input name="login"
type="text"><br>
    Senha:<input name="senha"
type="password"><br>
    <input type="submit">
  </fieldset>
</form>
```

- **Testem aí**
  - O que tem de novo?

## ➤ **Agrupar elementos**

### ➤ **fieldset ⇒ grupo de campos**

- Não quer dizer nada
- É um quadrado que aparece na página
  - Não faz diferença ao submeter para o servidor
- marcador **legend** é o que vai no texto do agrupador

Formulários

da aula anterior...

# Input $\Rightarrow$ Testem

```
<form>
  Nome:<input name="nome" type="text" value="nome"><br>
  Senha:<input name="senha" type="password" value="senha"><br>
  <input type="submit"><input type="reset">
</form>
```

Além do input

# Select

```
<form method="get"
action="teste.html" target="_blank">
    Nome:<input name="nome"
type="text">
    <br>
```

## ➤ Testem aí

- Select e seus options
- O que é o select?
- O que são os options?

```
    Estado: <select name="estado">
        <option value="rs"> RS
    </option>
        <option value="sc"> SC
    </option>
        <option value="pr"> PR
    </option>
        <option value="sp"> SP
    </option>
    </select>
    <input type="submit">
</form>
```

# Select

- **select ⇒ selecionar**
  - abre e fecha
  - selecionar entre as opções fornecidas
    - por padrão seleciona a primeira opção
    - **options ⇒ opções**
      - tem os valores fornecidos
  - parecido com o radio



# Select

- Testem aí
  - o que aconteceu?

```
<form method="get" action="teste.html"
target="_blank">
  Nome:<input name="nome" type="text">
  <br>
  Estado: <select name="estado">
    <option value="rs"> RS </option>
    <option value="sc"> SC </option>
    <option value="pr" selected> PR
  </option>
    <option value="sp"> SP </option>
  </select>
  <input type="submit">
</form>
```

# Atributo select

- **selected ⇒ selecionado**
  - alterado o option que está marcado
    - usado para colocar um como padrão que não é o primeiro

# Size

- Testem aí
  - o que aconteceu?

```
<form method="get" action="teste.html"
target="_blank">
  Nome:<input name="nome" type="text">
  <br>
  Estado: <select name="estado" size="2">
    <option value="rs"> RS </option>
    <option value="sc"> SC </option>
    <option value="pr" selected> PR
  </option>
    <option value="sp"> SP </option>
  </select>
  <input type="submit">
</form>
```

# atributo size

## ➤ **size ⇒ tamanho**

- altera a quantidade de options visualizadas por vez
- size="2" mostra duas opções
  - fica mais fácil de visualizar as opções

# multiple

- Testem aí
  - o que aconteceu?

```
<form method="get" action="teste.html"
target="_blank">
  Nome:<input name="nome" type="text">
  <br>
  Estado: <select name="estado" size="4"
multiple>
    <option value="rs"> RS </option>
    <option value="sc"> SC </option>
    <option value="pr" selected> PR
  </option>
    <option value="sp"> SP </option>
  </select>
  <input type="submit">
</form>
```

# atributo multiple

- **multiple** ⇒ **múltiplo**
  - deixa seleccionar várias opções
  - manda várias vezes o select

## textarea

- Testem aí
  - o que aconteceu?

```
<form method="get" action="teste.html"
target="_blank">
  Nome:<input name="nome" type="text">
  <br>
  Estado: <select name="estado" size="4"
multiple>
    <option value="rs"> RS </option>
    <option value="sc"> SC </option>
    <option value="pr" selected> PR
</option>
    <option value="sp"> SP </option>
  </select><br>
  <textarea name="mensagem">Busquem
conhecimento</textarea> <br>
  <input type="submit">
</form>
```

# marcador textarea

- **textarea ⇒ área de texto**
  - mostra um campo maior do que o input
  - para enviar texto mais longo
  - para um texto maior
    - o campo padrão fica um pouco pequeno



## textarea

- Testem aí
  - o que aconteceu?

```
<form method="get" action="teste.html"
target="_blank">
    Nome:<input name="nome" type="text">
    <br>
    Estado: <select name="estado" size="4"
multiple>
        <option value="rs"> RS </option>
        <option value="sc"> SC </option>
        <option value="pr" selected> PR
</option>
        <option value="sp"> SP </option>
    </select><br>
    <textarea name="mensagem" rows="50"
cols="20">Busquem conhecimento</textarea>
    <br>
    <input type="submit">
</form>
```

# atributos rows e cols

- **rows ⇒ linhas**

- define o número de linhas da caixa de texto que serão mostradas

- **cols ⇒ colunas**

- define o número de linhas da caixa de texto que serão mostradas
  - Aumentando esses números
    - aumenta a caixa de texto...

# inputs HTML5

## ➤ Até aqui vimos os inputs das versões mais antigas do html

➤ HTML5 veio com vários inputs novos nativos no navegador

➤ color

➤ date

➤ datetime-local

➤ email

➤ month

➤ number

➤ range

➤ search

➤ tel

➤ time

➤ url

➤ week

# inputs HTML5

- **Vários desses inputs eram implementados usando JS**
  - com esses inputs "novos"
  - não tem mais a necessidade da biblioteca/função JS
  - a funcionalidade é nativa do navegador
    - facilita bastante
    - padronização
    - carregamento mais rápido

<form>

Nome:<input name="login" type="text"><br>

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

Cor favorita: <input type="color" name="corFavorita">

<br>

Aniversário: <input type="date" name="aniversario"> <br>

Foto de perfil: <input type="file" name="foto"> <br>

Filhos: <input type="number" name="filhos"> <br>

Horário para contato: <input type="time" name="horario">

<br>

Página pessoal: <input type="url"

name="sitePessoal"><br>

<input type="submit">

</form>

# inputs HTML5

## ➤ email

- quase igual ao text
- informa o navegador que é um campo de email
- ajuda a autocompletar com dados passados
- não valida se é um email de fato

# inputs HTML5

## ➤ **color ⇒ cor**

- abre uma paleta de cores para o usuário escolher
- o valor do campo no submit é um hexadecimal da cor
  - padrão de cor

## ➤ **date ⇒ data**

- abre um calendário padrão para escolha da data
- evita que o usuário erre a data
- valida se é uma data correta

# inputs HTML5

## ➤ **file** ⇒ **arquivo**

- abre a tela para escolher um arquivo para enviar
- envia o arquivo ao servidor

## ➤ **number** ⇒ **número**

- obriga digitar somente números
- válida e tem como interagir através dos botões fornecidos



# inputs HTML5

## ➤ **time** ⇒ hora

- mesma coisa que o number só que para horas
- valida se a hora está correta (evita erros)

## ➤ **url**

- igual ao campo de texto
- verifica se o campo contém uma página válida
  - se não tem não manda para o servidor

# atributos dos inputs HTML5

Aniversário: `<input type="date" name="aniversario" max="2018-10-15" min="1940-01-01">` `<br>`

Filhos: `<input type="number" name="filhos" min="0" max="10" step="1" value="0">` `<br>`

# inputs HTML5

## ➤ **Cliquem no submeter/enviar sem preencher nenhum campo**

- Tá certo isso de  
enviar tudo em  
branco?

```
<form>
Nome:<input name="nome" type="text"><br>
E-mail: <input type="email" name="email"> <br>
Cor favorita: <input type="color" name="corFavorita">
<br>
Aniversário: <input type="date" name="aniversario">
<br>
Foto de perfil: <input type="file" name="foto"> <br>
Filhos: <input type="number" name="filhos"> <br>
Horário para contato:<input type="time" name="horario">
<br>
Página pessoal: <input type="url" name="sitePessoal">
<br>
<input type="submit">
</form>
```

# inputs HTML5

- **Cliquem no submeter/enviar sem preencher nenhum campo**
  - Tá certo isso?
  - atributo **required**
    - requerido ⇒ obrigatório
    - coloquem **required** dentro do marcador nome
      - tentem submeter com ele vazio
      - e aí!?

**MUITO  
OBRIGADO**

Próxima aula: mais HTML: atributos, formulários,  
introdução às folhas de estilo.