



Programação Web

Profº Iury Gomes de Oliveira

CONTEÚDO



- | | |
|------------------------|-----------------|
| 1. Objetivos do Curso; | 7. PHP; |
| 2. HTML; | 8. PHP+MYSQL; |
| 3. CSS; | 9. Referências. |
| 4. JavaScript; | |
| 5. Jquery; | |
| 6. AJAX; | |

1. OBJETIVOS



✓ Após esse curso o aluno será capaz de:

- Conhecer as principais linguagens de programação web utilizadas
- Produzir páginas web de forma consistente, de acordo com as exigências de padronização e seguindo a especificação Web Standard (especificado pela **W3C**);
- Criar scripts para aumentar a interatividade das páginas HTML, construindo páginas web com maior poder de interação no lado cliente;
- Iniciar o aluno na criação de Sistemas Web completos.

Video 1 - WEB

2. HTML



Basicamente, um documento HTML é composto por elementos hierarquicamente organizados. Para inserir um elemento em um documento HTML, devemos utilizar as tags correspondentes a esse elemento. As tags são definidas com parênteses angulares (< e >). Os elementos podem possuir atributos e conteúdo. Os atributos são formados por nome e valor. Normalmente, os valores dos atributos são definidos dentro de aspas dupla e o conteúdo dos elementos é um texto ou outros elementos.

2. HTML – ESTRUTURA BÁSICA



```
1 <! DOCTYPE html >
2 <html>
3
4   <head>
5     <title>
6       <!-- Título
7     </title>
8   </head>
9
10
11  <body >
12
13     <!-- Corpo do body -->
14
15  </body>
16
17 </html>
```

Para um navegador exibir corretamente uma página web, devemos informar explicitamente o tipo do documento. O tipo do documento é informado com a declaração **<!DOCTYPE>**.

2. HTML – ESTRUTURA BÁSICA



```
1 <! DOCTYPE html >
2 <html>
3
4   <head>
5     <title>
6       <!-- Título do site -->
7     </title>
8   </head>
9
10
11  <body >
12    <!-- Corpo do body -->
13
14
15  </body>
16
17 </html>
```

Essa tag diz ao navegador que o que estiver dentro dela deve ser encarado como elementos de um código HTML

2. HTML – ESTRUTURA BÁSICA



```
1 <! DOCTYPE html >
2 <html>
3
4 <head>
5   <title>
6     <!-- Título -->
7   </title>
8 </head>
9
10
11 <body >
12
13   <!-- Corpo -->
14
15 </body>
16
17 </html>
```

A principal função do elemento head é agrupar informações sobre o documento HTML (metainformação). São exemplos de metainformações: o encoding, a taxa de atualização, o autor, a descrição e as palavras chaves do documento HTML.

```
1 <head>
2   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
3   <meta http-equiv="refresh" content="30">
4   <meta name="author" content="Rafael Cosentino">
5   <meta name="description" content="Cursos de Java e .NET">
6   <meta name="keywords" content="curso java, curso C#">
7 </head>
```

2. HTML – ESTRUTURA BÁSICA



```
1 <! DOCTYPE html >
2 <html>
3
4   <head>
5       <title>
6           <!-- Título do site -->
7       </title>
8   </head>
9
10
11  <body >
12      <!-- Corpo do body -->
13
14  </body>
15
16
17 </html>
```


2. HTML – ESTRUTURA BÁSICA



```
1 <! DOCTYPE html >
2 <html>
3
4   <head>
5     <title>
6       <!-- Títu
7     </title>
8   </head>
9
10
11  <body >
12
13    <!-- Corpo do body -->
14
15  </body>
16
17 </html>
```

O conteúdo de uma página web deve ser definido no corpo do elemento body. Por exemplo, podemos inserir no corpo do body cabeçalhos, textos, listas, tabelas, entre outros componentes. E nesta parte que o conteúdo visível para o usuário ficará !

Podemos adicionar comentários em um documento HTML dentro das tags <!-- e -->.

2. HTML – NORMAL ELEMENT



Mais Sobre

De acordo com a especificação da linguagem HTML, alguns elementos são denominados **Normal Elements**. Esses elementos são abertos com uma tag e fechados com outra tag. Por exemplo, o elemento **p** é um **Normal Element**. Observe a utilização da tag de abertura e da tag de fechamento do elemento **p**.

```
1 <p>K02 - Desenvolvimento Web com HTML, CSS e JavaScript</p>
```

2. HTML – VOID ELEMENTS



Mais Sobre

Há também elementos denominados **Void Elements**. Esses elementos não possuem conteúdo. Na **sintaxe HTML**, esses elementos são abertos e fechados com apenas uma tag com ou sem o caractere `/`. Um exemplo de **Void Element** é o elemento `br`.

```
1 <p>
2   K01 - Lógica de Programação<br>
3   K02 - Desenvolvimento Web com HTML, CSS e JavaScript<br/>
4   K03 - SQL e Modelo Relacional
5 </p>
```

Os **Void Elements** são: `area`, `base`, `br`, `col`, `embed`, `hr`, `img`, `input`, `keygen`, `link`, `meta`, `param`, `source`, `track`, `wbr`.

Na **sintaxe XHTML**, os **Void Elements** podem ser abertos e fechados na mesma tag ou em tags separadas. Contudo, o caractere `/` é obrigatório.

```
1 <p>
2   K01 - Lógica de Programação<br/>
3   K02 - Desenvolvimento Web com HTML, CSS e JavaScript<br></br>
4   K03 - SQL e Modelo Relacional
5 </p>
```

2. HTML – TÍTULOS



Assim como em um livro, uma página web pode conter uma hierarquia de títulos para estabelecer uma divisão do seu conteúdo. Para inserir títulos em uma página web, devemos utilizar os elementos **h1, h2, h3, h4, h5 e h6**. Os sufixos numéricos de 1 a 6 indicam o **nível do título** dentro da hierarquia de títulos do documento.

2. HTML – TÍTULOS



```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>Exemplo de títulos</title>
6   </head>
7   <body>
8     <h1>Título 1</h1>
9     <h2>Título 2</h2>
10    <h3>Título 3</h3>
11    <h4>Título 4</h4>
12    <h5>Título 5</h5>
13    <h6>Título 6</h6>
14  </body>
15 </html>
```

Observem que os títulos não serão exibidos lado a lado e sim um embaixo do outro. Geralmente, os navegadores exibem os títulos como blocos. Por padrão, esses blocos ocupam todo o espaço horizontal do elemento onde os títulos estão contidos.

EXEMPLO

2. HTML – PARÁGRAFOS



Os parágrafos de uma página web são definidos através do elemento **p**.

Caso seja necessário forçar uma quebra de linha entre duas palavras contidas em um parágrafo, podemos utilizar o elemento **br**.

Quando um parágrafo contém palavras muito longas, os navegadores podem ter dificuldades para ajustar as quebras de linha. Podemos indicar explicitamente com o elemento **wbr**, como as palavras podem ser “quebradas”.

EXEMPLO

2. HTML – CHARACTER ENTITIES



Para inserir determinados caracteres em um documento HTML, devemos utilizar o código do caractere desejado. A utilização desses códigos evita problemas de encoding na exibição das páginas web. Você pode conhecer esses caracteres e os seus respectivos códigos através do seguinte endereço <http://dev.w3.org/html5/html-author/charref>. Esses caracteres são conhecidos como **Character Entities** ou **HTML Entities**

EXEMPLO

2. HTML – CHARACTER ENTITIES



```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>Exemplo de character entities</title>
6   </head>
7   <body>
8     <h1>Exemplo de character entities</h1>
9     <ul>
10      <li>&starf; &bigstar; &#x02605; &#9733;</li>
11      <li>&phone; &#x0260E; &#9742;</li>
12      <li>&female; &#x02640; &#9792;</li>
13      <li>&sung; &#x0266A; &#9834;</li>
14    </ul>
15  </body>
16 </html>
```

EXEMPLO

2. HTML – TEXTO PRÉ-FORMATADO



Como vimos, os navegadores desconsideram os espaços excedentes entre as palavras contidas em um documento HTML assim como desconsideram as quebras de linha. Contudo, é possível inserir texto formatado com quantos espaços e quebras de linha desejarmos através do elemento **pre**. O texto contido no conteúdo de um elemento **pre** é exibido com todos os espaços e quebras de linha inseridos no documento HTML.

EXEMPLO

2. HTML – TEXTO PRÉ-FORMATADO



```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>Exemplo de texto pré-formatado</title>
6   </head>
7   <body>
8     <h1>Exemplo de texto pré-formatado</h1>
9     <pre>
10 Os          espaços          excedentes   são considerados   pelos
11 navegadores. Assim como      as
12
13 quebras
14
15 de
16
17 linha. Observe também o tipo de      fonte   utilizada nesse texto.
18   </pre>
19 </body>
20 </html>
```

EXEMPLO

2. HTML – INSERÇÃO DE CODIGOS DE OUTRAS LINGUAGENS



Para adicionar códigos em um documento HTML, devemos utilizar o elemento **code**. Normalmente, esse elemento é utilizado para definir códigos escritos em alguma linguagem de programação.

EXEMPLO

2. HTML – TEXTO EM ITÁLICO



Para adicionarmos texto em itálico devemos utilizar a tag **i**

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>Exemplo de utilização do elemento i</title>
6   </head>
7   <body>
8     <h1>Exemplo de utilização do elemento i</h1>
9     <ul>
10      <li>Porquinho-da-índia ou <i>Cavia porcellus</i></li>
11      <li><i>Backup</i>(cópia de segurança)</li>
12      <li><i>shoot the breeze</i>(bater papo ou jogar conversa fora)</li>
13      <li><i>Moskvá</i>(transliteração da palavra Moscou em russo)</li>
14      <li><i>Se não sabes, aprende; se já sabes, ensina.</i> (Confúcio)</li>
15    </ul>
16  </body>
17 </html>
```

2. HTML – TEXTO EM ITÁLICO



Para adicionarmos texto em itálico devemos utilizar a tag **i**

A screenshot of a web browser window. The address bar shows the URL 'localhost/~keizo/html/public_html/i.html'. The page content features a large heading 'Exemplo de utilização do elemento i' followed by a bulleted list of examples of italicized text.

Exemplo de utilização do elemento i

- Porquinho-da-índia ou *Cavia porcellus*
- *Backup*(cópia de segurança)
- *shoot the breeze*(bater papo ou jogar conversa fora)
- *Moskvá*(transliteração da palavra Moscou em russo)
- *Se não sabes, aprende; se já sabes, ensina.* (Confúcio)

2. HTML – TEXTO EM NEGRITO



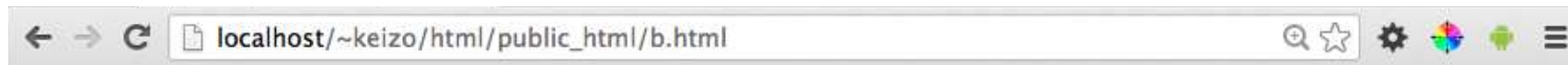
Para adicionarmos texto em negrito podemos usar a tag **** ou ****

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>Exemplo de utilização do elemento b</title>
6   </head>
7   <body>
8     <h1>Exemplo de utilização do elemento b</h1>
9     <p>
10      Atualmente, praticamente todos os <b>sistemas corporativos</b>
11      possuem <b>interfaces web</b>. Para quem deseja atuar no mercado
12      de <b>desenvolvimento de software</b>, é obrigatório o conhecimento
13      das linguagens: <b>HTML</b>, <b>CSS</b> e <b>JavaScript</b>.
14    </p>
15  </body>
16 </html>
```

2. HTML – TEXTO EM NEGRITO



Para adicionarmos texto em negrito podemos usar a tag **** ou ****



Exemplo de utilização do elemento b

Atualmente, praticamente todos os **sistemas corporativos** possuem **interfaces web**. Para quem deseja atuar no mercado de **desenvolvimento de software**, é obrigatório o conhecimento das linguagens: **HTML**, **CSS** e **JavaScript**.

2. HTML – TEXTO EM NEGRITO



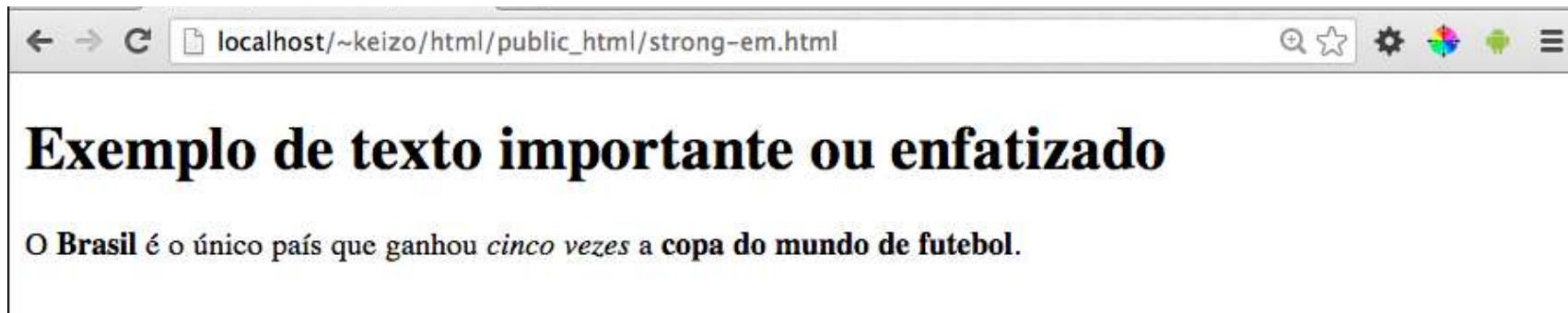
Para adicionarmos texto em negrito podemos usar a tag **** ou ****

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>Exemplo de texto importante ou enfatizado</title>
6   </head>
7   <body>
8     <h1>Exemplo de texto importante ou enfatizado</h1>
9     <p>
10       0 <strong>Brasil</strong> é o único
11       país que ganhou <em>cinco vezes</em>
12       a <strong>copa do mundo de futebol</strong>.
13     </p>
14   </body>
15 </html>
```


2. HTML – TEXTO EM NEGRITO



Para adicionarmos texto em negrito podemos usar a tag **** ou ****



2. HTML – TEXTO COM ALTERAÇÕES



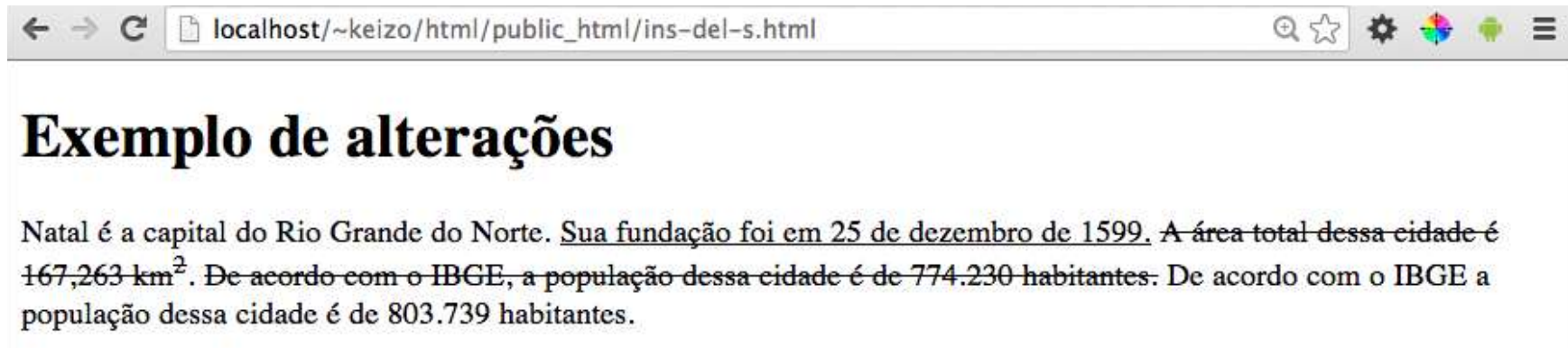
Em alguns casos, é importante informar que o texto de um documento HTML sofreu alterações. Um texto que foi acrescentado em um documento HTML deve ser definido com o elemento **ins**. Um texto que não faz mais parte do documento deve ser definido com o elemento **del**. Um texto que deixou de ser correto, preciso ou relevante deve ser definido com o elemento **s**.

2. HTML – TEXTO COM ALTERAÇÕES



```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>Exemplo de alterações</title>
6   </head>
7   <body>
8     <h1>Exemplo de alterações</h1>
9     <p>
10      Natal é a capital do Rio Grande do Norte. <ins>Sua fundação foi
11      em 25 de dezembro de 1599.</ins> <del>A área total dessa cidade é
12      167,263 km<sup>2</sup></del>. <s>De acordo com o IBGE, a população
13      dessa cidade é de 774.230 habitantes.</s> De acordo com o IBGE a
14      população dessa cidade é de 803.739 habitantes.
15    </p>
16  </body>
17 </html>
```

2. HTML – TEXTO COM ALTERAÇÕES



2. HTML – TEXTO SUBSCRITO E SOBRESCRITO



Podemos definir textos subscrito ou sobrescrito com os elementos **sub** e **sup** respectivamente.

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>Exemplo de texto subscrito ou sobrescrito</title>
6   </head>
7   <body>
8     <h1>Exemplo de texto subscrito ou sobrescrito</h1>
9     <p>
10       A cidade de São Paulo possui uma área de 1.523 km<sup>2</sup>.
11       Em 2011, São Paulo emitiu 16,430 milhões de toneladas de
12       CO<sub>2</sub>.
13     </p>
14   </body>
15 </html>
```

2. HTML – TEXTO SUBSCRITO E SOBRESCRITO



Podemos definir textos subscrito ou sobrescrito com os elementos **sub** e **sup** respectivamente.



2. HTML – TEXTO COM EFEITO DE MARCA TEXTO



O elemento **mark** permite que determinados trechos de um texto sejam marcados.

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>Exemplo de texto marcado</title>
6   </head>
7   <body>
8     <h1>Exemplo de texto marcado</h1>
9     <p>
10       <mark>Java</mark> e <mark>C#</mark> são linguagens de programação
11       <mark>orientadas a objeto</mark>.
12     </p>
13   </body>
14 </html>
```

2. HTML – TEXTO COM EFEITO DE MARCA TEXTO




O elemento **mark** permite que determinados trechos de um texto sejam marcados.



2. HTML – LISTAS



A linguagem HTML define três tipos de listas:

- a) *Lista com descrições*: Para criar uma lista de descrições, devemos utilizar o elemento **dl**. Essas listas são formadas por termos ou nomes e as suas respectivas descrições. Os termos ou nomes são definidos com o elemento **dt**. As descrições são definidas com o elemento **dd**.  **EXEMPLO**
- b) *Lista com ordem*: Para criar uma lista com ordem, devemos utilizar o elemento **ol**. Os itens de uma lista com ordem são definidos com o elemento **li**
- c) *Lista sem ordem*: Para criar uma lista sem ordem, devemos utilizar o elemento **ul**. Os itens de uma lista sem ordem são definidos com o elemento **li**.

2. HTML – LISTA COM ORDEM



```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>Exemplo de lista com ordem</title>
6   </head>
7   <body>
8     <h1>Macarrão instantâneo - K19 Receitas</h1>
9     <h2>Modo de preparo</h2>
10
11     <ol>
12       <li>Ferver 600ml de água em uma panela.</li>
13       <li>Retirar o macarrão do pacote.</li>
14       <li>Colocar o macarrão na panela no fogo baixo.</li>
15       <li>Cozinhar o macarrão por 3min.</li>
16       <li>Temperar a gosto.</li>
17     </ol>
18   </body>
19 </html>
```

2. HTML – LISTA COM ORDEM



 localhost/~keizo/html/public_html/ol-li.html 

Macarrão instantâneo - K19 Receitas

Modo de preparo

1. Ferver 600ml de água em uma panela.
2. Retirar o macarrão do pacote.
3. Colocar o macarrão na panela no fogo baixo.
4. Cozinhar o macarrão por 3min.
5. Temperar a gosto.

2. HTML – LISTA SEM ORDEM



```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>Exemplo de lista sem ordem</title>
6   </head>
7   <body>
8     <h1>K02 - Desenvolvimento Web com HTML, CSS e JavaScript</h1>
9     <h2>Pré-requisitos</h2>
10
11     <ul>
12       <li>Conhecimento de algum sistema operacional (Windows/Linux/MacOS X)</li>
13       <li>Lógica de programação</li>
14     </ul>
15   </body>
16 </html>
```

2. HTML – LISTA ANINHADA



Uma lista pode ser definida dentro de outra lista. Quando listas sem ordem são aninhadas, normalmente, os navegadores alternam o marcadores dos itens.

EXEMPLO

2. HTML – LINKS



Normalmente, um site é formado por um conjunto de páginas que estão interligadas de alguma forma. Para permitir que um usuário navegue de uma página para outra, devemos utilizar os links.

Basicamente, um link faz a ligação de uma página para outra do mesmo site (link interno) ou para uma página de outro site (link externo).

Para criarmos um link, devemos utilizar o elemento **a**. Esse elemento possui um atributo chamado **href**. O valor desse atributo indica o caminho **relativo ou absoluto** de uma outra página.

2. HTML – LINKS



```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>Exemplo de links</title>
6   </head>
7   <body>
8     <ul>
9       <li>
10        <a href="pagina2.html">Exemplo de link com caminho relativo</a>
11      </li>
12      <li>
13        <a href="outros/pagina3.html">Outro exemplo de link com caminho relativo</a>
14      </li>
15      <li>
16        <a href="http://www.k19.com.br">Exemplo de link com caminho absoluto</a>
17      </li>
18    </ul>
19  </body>
20 </html>
```

2. HTML – LINKS



- [Exemplo de link com caminho relativo](#)
- [Outro exemplo de link com caminho relativo](#)
- [Exemplo de link com caminho absoluto](#)

2. HTML – LINKS



Além do atributo **href**, podemos utilizar o atributo **target** para informar onde o destino de um link deve ser aberto.

- **_blank**: Abre o destino do link em uma nova janela ou aba.
- **_self**: Abre o destino do link na mesma janela ou no mesmo frame que exibe o documento que contém o link.
- **_parent**: Abre o destino do link na janela ou no frame onde está contido o frame que exibe o documento que contém o link.
- **_top**: Abre o destino do link na janela que é “ancestral” do frame que exibe o documento que contém o link.

2. HTML – LINKS



Além do atributo **href**, podemos utilizar o atributo **target** para informar onde o destino de um link deve ser aberto.

- **_blank**: Abre o destino do link em uma nova janela ou aba.
- **_self**: Abre o destino do link na mesma janela ou no mesmo frame que exibe o documento que contém o link.
- **_parent**: Abre o destino do link na janela ou no frame onde está contido o frame que exibe o documento que contém o link.
- **_top**: Abre o destino do link na janela que é “ancestral” do frame que exibe o documento que contém o link.

2. HTML – LINKS



```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>Exemplo de uso da tag a com o atributo target</title>
6   </head>
7   <body>
8     <ul>
9       <li>
10        <a href="pagina1.html" target="_blank">Abre em outra janela ou aba</a>
11      </li>
12      <li>
13        <a href="pagina2.html" target="_self">Abre na mesma janela ou aba</a>
14      </li>
15      <li>
16        <a href="pagina3.html">Abre na mesma janela ou aba</a>
17      </li>
18    </ul>
19  </body>
20 </html>
```

2. HTML – ÂNCORAS



Além de criar links para outras páginas, podemos criar um link para uma determinada seção de um documento HTML. Esse recurso é chamado de **ancoragem**. O primeiro passo para utilizar esse recurso é identificar a seção que será o destino desse link. Essa identificação é realizada com o atributo **id**. O **id** é um **atributo global**, ou seja, todos os elementos possuem esse atributo. O segundo passo é criar os links utilizando os identificadores das seções de acordo com a sintaxe do exemplo a seguir. Observe a utilização do caractere #.

EXEMPLO

2. HTML – IMAGENS



Os sites seriam muito entediantes se não fosse possível adicionar imagens ao conteúdo das páginas. Podemos adicionar imagens em documento HTML com o elemento **img**. Esse elemento possui um atributo chamado **src**. Esse atributo indica o caminho absoluto ou relativo da imagem que queremos adicionar.

No elemento **img** pode-se adicionar o atributo **alt**. Esse atributo define um texto alternativo que pode ser utilizado, por exemplo, se houver um problema ao carregar a imagem ou por softwares de leitura de tela.

2. HTML – URL ABSOLUTA E RELATIVA



Os links e as imagens podem ser adicionados em um documento HTML com URLs **absolutas** ou **relativas**.

Qual a diferença ?

2. HTML – URL ABSOLUTA E RELATIVA



URL RELATIVA

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>URLs absolutas e relativas</title>
6   </head>
7   <body>
8     
9     <a href="paginas/cursos.html">Cursos</a>
10  </body>
11 </html>
```

A imagem deve estar na mesma pasta de arquivos em que o arquivo do site está presente

2. HTML – URL ABSOLUTA E RELATIVA



URL ABSOLUTA

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5     <title>URLs absolutas e relativas</title>
6   </head>
7   <body>
8     
9     <a href="http://www.k19.com.br/paginas/cursos.html">Cursos</a>
10  </body>
11 </html>
```

2. HTML – TABELAS



Para exibir dados de forma tabular em páginas web, usa-se o elemento **table**. Esse elemento permite apresentar dados de forma tabular. As linhas de uma tabela são definidas com o elemento **tr**, as células de títulos com o elemento **th** e as células de dados com o elemento **td**. Os elementos **th** e **td** possuem um atributo chamado **colspan** e outro chamado **rowspan**. O atributo **border** permite ligar as bordas tabela.

EXEMPLO

2. HTML – EXERCÍCIOS



2. Crie um site que contenha:

- a) Uma lista com descrições;
- b) Uma lista ordenada;
- c) Uma lista sem ordem;
- d) Crie um segundo site, e faça com que o 1º site criado nos exercícios anteriores faça referência ao 2º site, através de links. De igual modo faça com que o 2º site também referencie o 1º site;
- e) A utilização do recursos de ancoragem
- f) A utilização do recurso de tabela

2. HTML – EXERCÍCIOS



2. Crie um site que contenha:

- g) Realize uma pesquisa e descubra a utilidade dos atributos **colspan**, **rowspan**, **thead**, **tbody**, **tfoot**, **align**, **bgcolor** combinados com os comandos de criação de tabelas em html

2. HTML – FORMULÁRIOS



Atualmente, praticamente quase todas as aplicações web necessitam obter informações enviadas pelos usuários. Por exemplo, considere uma empresa que deseja receber os pedidos dos seus clientes através do seu site. O site dessa empresa precisa oferecer alguma forma de interação que possibilite o recebimento de dados fornecidos pelos usuários.

Para tornar os sites e as aplicações web mais interativos, podemos utilizar **formulários**. Através dos formulários, os usuários podem enviar informações aos servidores web.

2. HTML – FORMULÁRIOS



Para criar um formulário, devemos utilizar o elemento **form**. Esse elemento possui um atributo chamado **action**. O valor desse atributo indica para qual endereço os dados do formulário serão enviados.

```
1 <form action="pagina.html">
2 ...
3 </form>
```

2. HTML – FORMULÁRIOS



Os formulários são compostos por **caixas de texto**, **checkboxes**, **radios**, **caixas de seleção**, **botões**, entre outros componentes.

Caixa de texto	Caixa de senha	Caixa de data																																										
Nome: <input type="text" value="Jonas"/>	Senha: <input type="password" value="....."/>	Nascimento: <input type="text" value="09/18/2013"/>																																										
		<div>September 2013 ▾</div> <table><thead><tr><th>Sun</th><th>Mon</th><th>Tue</th><th>Wed</th><th>Thu</th><th>Fri</th><th>Sat</th></tr></thead><tbody><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td></tr><tr><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td></tr><tr><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td></tr><tr><td>29</td><td>30</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></tbody></table>	Sun	Mon	Tue	Wed	Thu	Fri	Sat	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	5
Sun	Mon	Tue	Wed	Thu	Fri	Sat																																						
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	5																																						

2. HTML – FORMULÁRIOS



Os formulários são compostos por **caixas de texto**, **checkboxes**, **radios**, **caixas de seleção**, **botões**, entre outros componentes.

Checkboxes	Radio	Botões
Java <input checked="" type="checkbox"/> C# <input checked="" type="checkbox"/> PHP <input type="checkbox"/>	Manhã <input type="radio"/> Tarde <input checked="" type="radio"/> Noite <input type="radio"/>	<input type="button" value="ok"/> <input type="button" value="cancelar"/>

2. HTML – FORMULÁRIOS – CAIXAS DE TEXTO



Geralmente, os formulários possuem uma ou mais caixas de texto. As caixas de texto são adicionadas nos documentos HTML através do elemento **input**. Esse elemento possui um atributo chamado **type**. Para definir uma caixa de texto, o valor do atributo **type** deve ser **text**.

Considere um formulário com diversas caixas de texto. Quando as informações preenchidas nesse formulário chegarem ao Servidor Web, certamente, ele precisará saber o que foi preenchido em cada caixa. Por isso, é necessário identificar esses dados. O atributo **name** do elemento **input** é utilizado para resolver esse problema.

2. HTML – FORMULÁRIOS – CAIXAS DE TEXTO



```
1 <input type="text" name="endereco">  
2 <input type="text" name="cidade">
```

2. HTML – FORMULÁRIOS – PARÂMETROS GET E POST



Ao desenvolver um site dinâmico, existe a necessidade de passar alguns valores de uma página para a outra, para realizar operações como consultas e inserções no banco, autenticação de usuários, etc.

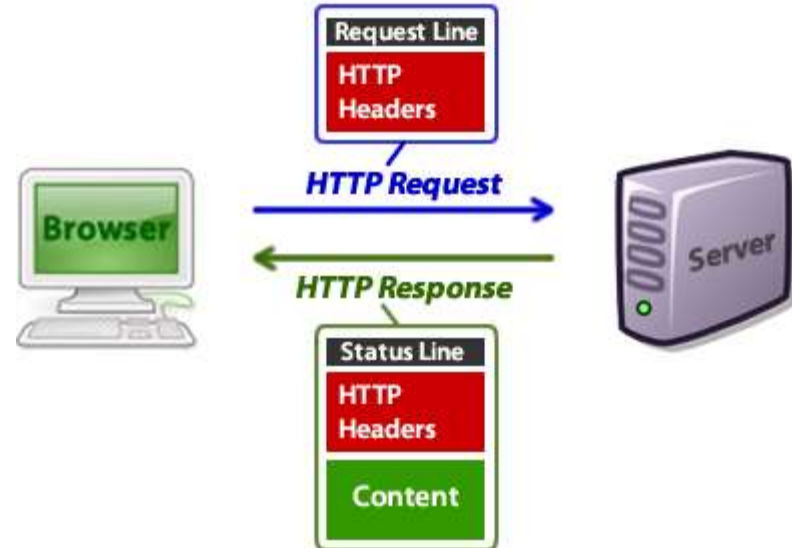
Os principais métodos de comunicação HTTP são: **GET** e **POST**.

Visibilidade – A grande diferença entre os métodos GET e POST provavelmente é a visibilidade. Uma requisição GET é enviada como string anexada a URL, enquanto que a requisição POST é encapsulada junto ao corpo da requisição HTTP e não pode ser vista. Em termos de segurança o método POST é mais recomendado.

2. HTML – FORMULÁRIOS – PARÂMETROS GET E POST



Tamanho – Como a requisição GET é feita via URL, obviamente há uma limitação no tamanho da mensagem enviada. A string não pode conter mais que 255 caracteres(embora exista diferenças entre navegadores, mas em geral o limite é 255). Já na requisição POST não há limitações de comprimento da mensagem, já que a mesma é enviada no corpo da requisição HTTP.



2. HTML – FORMULÁRIOS – PARÂMETROS GET E POST



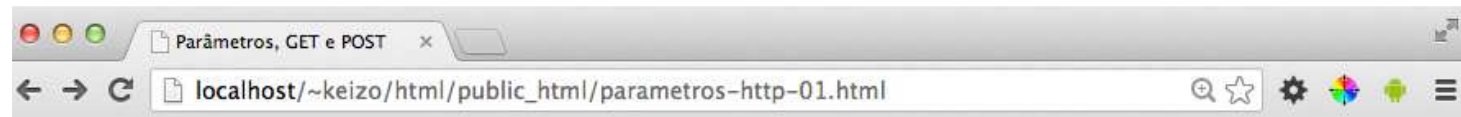
Performance – A requisição GET é relativamente mais rápida, já que ela é mais simples. Na requisição POST há uma perda de tempo no encapsulamento da mensagem.

Tipos – Já que GET é enviado via URL, então nós sabemos que ela só transporta textos. A requisição POST não tem restrições, pode transportar tanto texto, como dados binários.

Favoritos/Bookmarks – Por se tratar apenas de uma URL, a requisição GET pode ser armazenada em cache, ou em um sistema de bookmark(favoritos). A mesma coisa não é possível para requisições POST.

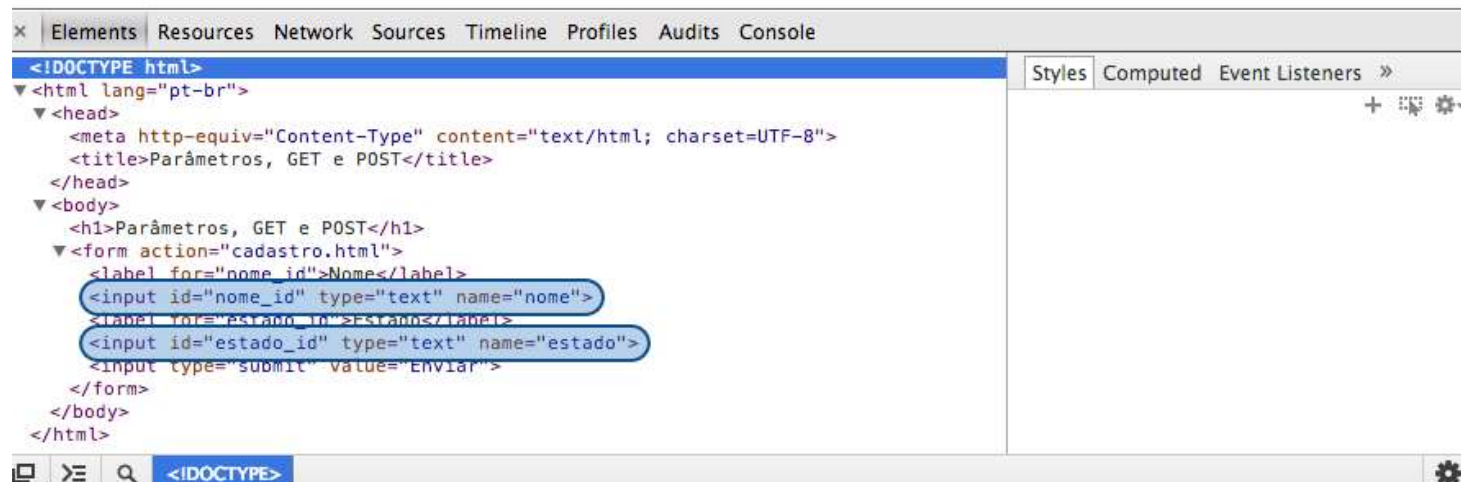
Método HTML Padrão – GET é o método HTML padrão. Para submeter um formulário HTML usando POST é preciso especificar no atributo “method” o valor “POST”.

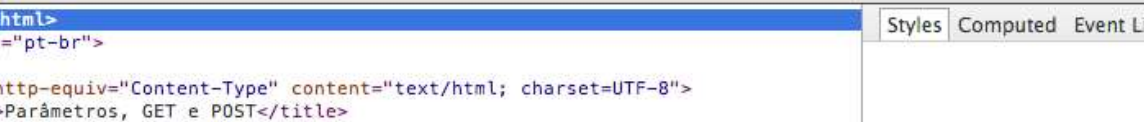
2. HTML – FORMULÁRIOS – PARÂMETROS GET E POST



Parâmetros, GET e POST

Nome Estado





```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Parâmetros, GET e POST</title>
  </head>
  <body>
    <h1>Parâmetros, GET e POST</h1>
    <form action="cadastro.html">
      <label for="nome_id">Nome</label>
      <input id="nome_id" type="text" name="nome">
      <label for="estado_id">Estado</label>
      <input id="estado_id" type="text" name="estado">
      <input type="submit" value="Enviar">
    </form>
  </body>
</html>
```

2. HTML – FORMULÁRIOS – PARÂMETROS GET E POST



GET:

Parâmetros, GET e POST

Formulário enviado com sucesso.

Elements Resources Network Sources Timeline Profiles Audits Console

Name Path

cadastro.html?nome=Jo...
/~keizo/html/public_html

Request URL: http://localhost/~keizo/html/public_html/cadastro.html?nome=Jonas+Hirata&estado=SP
Request Method: GET
Status Code: 200 OK

Request Headers

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
Connection: keep-alive
Cookie: PHPSESSID=f9mmsak4c1ldnud8lo2adidhb4; JSESSIONID=682ae7124aa6f4dbb8d7b237cf3a; treeForm_tree-hi=
Host: localhost
Referer: http://localhost/~keizo/html/public_html/parametros-http-01.html
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/30.0.1599.47 Safari/537.36

Query String Parameters

nome: Jonas Hirata
estado: SP

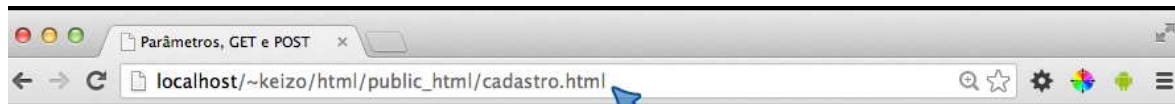
1 requests | 631 B transferred

Documents Stylesheets Images Scripts XHR Fonts WebSockets Other

2. HTML – FORMULÁRIOS – PARÂMETROS GET E POST

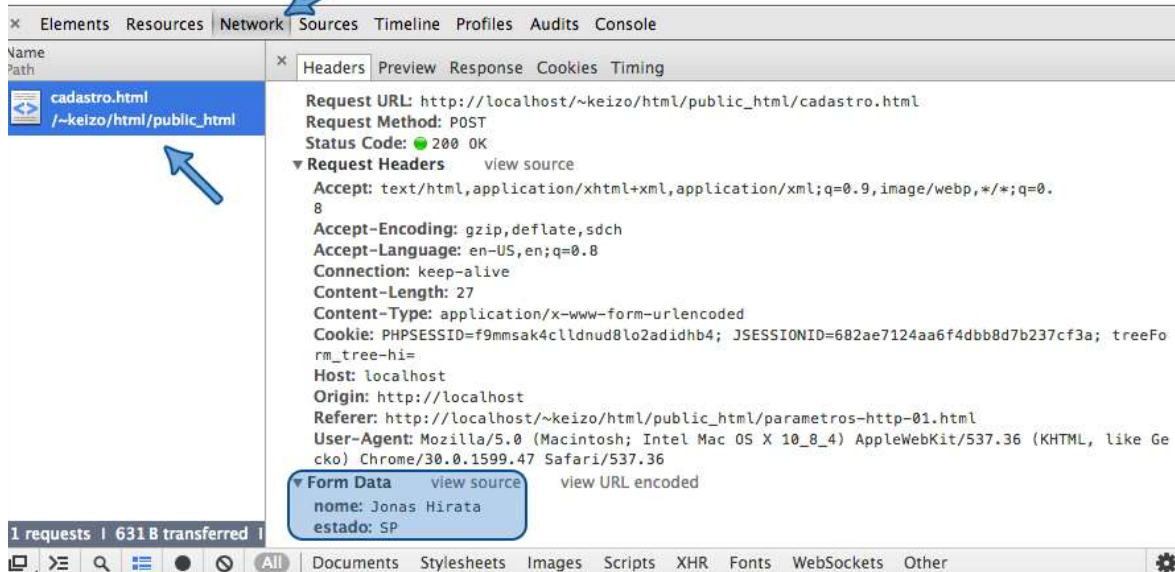


POST:



Parâmetros, GET e POST

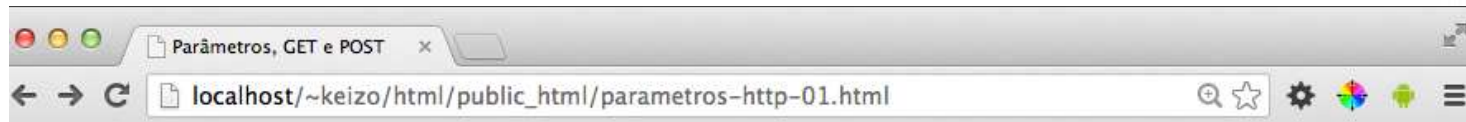
Formulário enviado com sucesso.



2. HTML – FORMULÁRIOS – PARÂMETROS GET E POST



POST:



Parâmetros, GET e POST

Nome Estado



2. HTML – FORMULÁRIOS – RÓTULOS



Nos formulários, os rótulos são fundamentais para informar aos usuários quais dados devem ser preenchidos. Para adicionar um rótulo, devemos utilizar o elemento **label**. Os textos dos rótulos são definidos no conteúdo desse elemento.

Para melhorar a acessibilidade dos documentos HTML, os rótulos devem ser explicitamente associados aos campos dos formulários. Para estabelecer esse vínculo, o primeiro passo é identificar os campos através do atributo **id**. O segundo passo é definir o atributo **for** do elemento **label** com o identificador do campo correspondente ao rótulo.

2. HTML – FORMULÁRIOS – RÓTULOS



```
1 <label>Nome: </label>  
2 <input type="text" name="nome">
```

```
1 <label for="nome_id">Nome: </label>  
2 <input type="text" name="nome" id="nome_id">
```

Nome:

2. HTML – FORMULÁRIOS – BOTÕES DE ENVIAR



Para adicionar um botão de submit em um formulário, podemos utilizar o elemento **input** com **type** igual a **submit**. Esse tipo de botão envia os dados do formulário para o Web Server. Os textos desses botões são definidos com o atributo **value**.

```
1 <input id="botao_id" type="submit" value="enviar">
```



2. HTML – FORMULÁRIOS – BOTÕES DE ENVIAR



Outra forma de adicionar um botão de submit em um documento HTML é utilizar o elemento **button** com **type** igual a **submit**. Diferentemente do elemento **input**, o elemento **button** permite a criação de botões com imagens além de texto.

```
1 <button id="botao_id" type="submit">
2   Enviar
3   
4 </button>
```



2. HTML – FORMULÁRIOS – BOTÕES DE ENVIAR



Uma imagem pode funcionar como um botão de submit. Para isso, devemos utilizar o elemento **input** com **type** igual a **image**. O caminho absoluto ou relativo da imagem que será utilizada deve ser definida como atributo **src**. Um texto alternativo deve ser definido com o atributo **alt**. Esse texto pode ser utilizado caso ocorra algum problema no carregamento da imagem.

```
1 <form action="parametros.php" method="post" enctype="multipart/form-data">
2   <input id="botao_id" alt="enviar" type="image" src="submit.png">
3 </form>
```

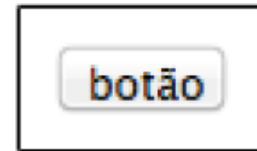


2. HTML – FORMULÁRIOS – BOTÕES GENÉRICOS



Para adicionar um botão genérico em um formulário, podemos utilizar o elemento **input** com **type** igual a **button**. As ações desse tipo de componente são definidas com JavaScript. Os textos desses botões são definidos com o atributo **value**.

```
1 <input id="botao_id" type="button" value="botão">
```



2. HTML – FORMULÁRIOS – BOTÕES GENÉRICOS



Outra forma de adicionar um botão genérico em um documento HTML é utilizar o elemento **button** com **type** igual a **button**. Diferentemente do elemento **input**, o elemento **button** permite a criação de botões com imagens além de texto.

```
1 <button id="botao_id" type="button">
2   Botão genérico
3   
4 </button>
```

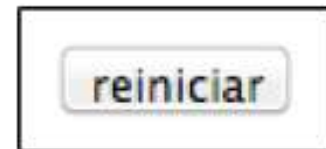


2. HTML – FORMULÁRIOS – BOTÕES RESET



Para adicionar um botão de reset em um formulário, podemos utilizar o elemento **input** com **type** igual a **reset**. Esse tipo de botão reinicia os dados do formulário. Os textos desses botões são definidos com o atributo **value**.

```
1 <input id="botao_id" type="reset" value="reiniciar">
```



2. HTML – FORMULÁRIOS – BOTÕES RESET



Outra forma de adicionar um botão de reset em um documento HTML é utilizar o elemento **button** com **type** igual a **reset**. Diferentemente do elemento **input**, o elemento **button** permite a criação de botões com imagens além de texto.

```
1 <button id="botao_id" type="reset">
2   Botão de reset
3   
4 </button>
```

2. HTML – FORMULÁRIOS – BOTÕES DE UPLOAD



Para adicionar um botão de upload em um formulário, podemos utilizar o elemento **input** com **type** igual a **file**. Esse tipo de botão permite selecionar um arquivo para um eventual upload. O formulário que contém esse botão deve possuir o atributo **enctype** com o valor **multipart/formdata**.

```
1 <input id="botao_id" name="file" type="file">
```



2. HTML – FORMULÁRIOS – CAIXAS DE BUSCA



Assim como as caixas de texto, as caixas de busca são adicionadas nos formulário com o elemento **input**. A diferença é que o valor do atributo **type** deve ser **search** ao invés de **text**.

As caixas de busca devem ser utilizadas para coletar palavras chave que serão utilizadas em algum tipo de pesquisa. A princípio não há nenhuma diferença prática entre as caixas de texto e as caixas de busca. Contudo, essa diferenciação adiciona valor semântico aos documentos HTML e possibilita, por exemplo, que os navegadores diferenciem visualmente esses dois tipos de caixas.

2. HTML – FORMULÁRIOS – CAIXAS DE BUSCA



```
<input id="keywords_id" name="keywords" type="search">
```

Busca:

2. HTML – FORMULÁRIOS – CAIXAS DE NÚMEROS



Para coletar dados numéricos, podemos utilizar caixas específicas para números. No HTML5, há dois tipos de caixas para esse propósito. Os dois são definidos com o elemento **input**. O valor do atributo **type** é **number** para o primeiro tipo e **range** para o segundo tipo.

Esses dois tipos de componentes devem ser utilizados para coletar valores de sequências numéricas pré-definidas. A principal diferença entre eles é que o primeiro (**type=number**) deve oferecer um mecanismo preciso para os usuários selecionarem o valor desejado enquanto o segundo (**type=range**) não possui essa obrigação. A imagem a seguir mostra uma possível forma dos navegadores exibirem essas caixas.

2. HTML – FORMULÁRIOS – CAIXAS DE NÚMEROS



```
1 <input id="numero1_id" name="numero1" type="number">  
2 <input id="numero2_id" name="numero2" type="range">
```

Idade:

Quantidade:

2. HTML – FORMULÁRIOS – CAIXAS DE NÚMEROS



Para definir a sequência dos números que podem ser selecionados pelos usuários, podemos utilizar os atributos **min**, **max** e **step**. Por exemplo, para coletar um número da sequência {0; 0,2; 0,4; 0,6; 0,8; 1}, os valores dos atributos **min**, **max** e **step** devem ser **0**, **1** e **0.2** respectivamente.

```
<input id="numero1_id" type="number" name="numero1" min="0" max="1" step="0.2">  
<input id="numero2_id" type="range" name="numero2" min="0" max="1" step="0.2">
```

2. HTML – FORMULÁRIOS – CAIXAS DE EMAIL, TEL E URL



No HTML5 foram definidas caixas de entradas específicas para emails, telefones e urls. Essas caixas são adicionadas com o elemento **input**. O valor do atributo **type** deve ser **email**, **tel** e **url** para emails, telefones e urls respectivamente.

A usabilidade das páginas web melhora com a utilização dessas caixas. Por exemplo, a configuração do teclado dos celulares ou tablets pode ser alterada de acordo com o tipo de caixa de entrada.

Nas caixas de email, o caractere “@” pode ser adicionado ao teclado. Nas caixas de telefone, o teclado não precisa conter as letras do alfabeto. Nas caixas de url, teclas especiais como “.com” ou “www” podem ser adicionadas ao teclado.

2. HTML – FORMULÁRIOS – CAIXAS DE EMAIL, TEL E URL



```
<input id="email_id" name="email" type="email">  
<input id="telefone_id" name="telefone" type="tel">  
<input id="url_id" name="url" type="url">
```

2. HTML – FORMULÁRIOS – CAIXAS DE EMAIL, TEL E URL

Three side-by-side browser screenshots showing a web form with the following fields: E-mail, Telephone, and Site. Each field has a text input box. Below the fields is a button labeled 'Enviar'. The browser address bar shows '192.168.1.118/~keizo/html/pub'. The form is displayed on a mobile device with a QWERTY keyboard. The keyboard has a blue 'Ir' button. The first two screenshots show the form with the 'Enviar' button highlighted. The third screenshot shows the form with the 'Enviar' button highlighted and a blue 'Avançar' button at the bottom right of the keyboard.

2. HTML – FORMULÁRIOS – CAIXAS DE EMAIL, TEL E URL



Diversos tipos de caixas de entrada para coletar datas e horas foram adicionados no HTML5. Todas essas caixas são adicionadas com o elemento **input** e o valor do atributo **type** desse elemento assumirá um dos valores listados a seguir.

- **date**: Utilizado para coletar data (dia, mês e ano) sem fuso horário.
- **datetime**: Utilizado para coletar data (dia, mês e ano) e hora (hora, minuto, segundo e fração de segundo) com fuso horário em UTC.
- **datetime-local**: Utilizado para coletar data (dia, mês e ano) e hora (hora, minuto, segundo e fração de segundo) sem fuso horário.

2. HTML – FORMULÁRIOS – CAIXAS DE EMAIL, TEL E URL



- **month**: Utilizado para coletar data composta por mês e ano sem fuso horário.
- **time**: Utilizado para coletar hora (hora, minuto, segundo e fração de segundo) sem fuso horário.
- **week**: Utilizado para coletar data composta por semana e ano sem fuso horário.

A criação desses componentes permite que os navegadores melhorem a usabilidade das páginas web. A forma de exibição das caixas de datas e horas pode facilitar o processo de preenchimento dos formulários. Inclusive, os navegadores podem exibir esses componentes de formas diferentes de acordo com o dispositivo (computador, celular, tablet, entre outros).

2. HTML – FORMULÁRIOS – CAIXAS DE EMAIL, TEL E URL



```
<input id="data_id" name="data" type="date">  
<input id="data_hora_fuso_id" name="data-hora-fuso" type="datetime">  
<input id="data_hora_id" name="data-hora" type="datetime-local">  
<input id="mes_id" name="mes" type="month">  
<input id="hora_id" name="hora" type="time">  
<input id="semana_id" name="semana" type="week">
```

Descubra o que acontece no site quando os códigos são digitados acima

2. HTML – FORMULÁRIOS – CAIXAS DE SENHA



As senhas devem ser coletadas com caixas específicas para esse tipo de informação. Para adicionar uma caixa de senha em um formulário, devemos utilizar o elemento **input** com o valor **password** para o atributo **type**. Normalmente, os navegadores utilizam símbolos como o asterisco ou o círculo para omitir o conteúdo das caixas de senha.

```
<input id="senha_id" name="senha" type="password">
```

Senha:

2. HTML – FORMULÁRIOS – CAIXAS DE TEXTO LONGO



Para coletar um texto com várias linhas, podemos utilizar o elemento **textarea**. A quantidade de linhas de um **textarea** é definida com o atributo **rows** e a quantidade de colunas com o atributo **cols**.

Esse elemento também possui o atributo **name** que funciona como no elemento **input**.

Podemos definir o limite de caracteres que podem ser inseridos no conteúdo do elemento **textarea** através do atributo **maxlength**.

2. HTML – FORMULÁRIOS – CAIXAS DE TEXTO LONGO



```
<textarea id="mensagem_id" name="mensagem" rows="5" cols="50" maxlength="140">  
</textarea>
```

Texto:

2. HTML – FORMULÁRIOS – CHECKBOXES E RADIOS



Para adicionar um checkbox em um formulário, devemos utilizar o elemento **input** com **type** igual a **checkbox**. Ao utilizar esse componente, é importante definir um valor para o atributo **value**. No envio do formulário, esse valor é transmitido ao Web Server se o checkbox correspondente estiver marcado.

Eventualmente é interessante agrupar um determinado conjunto de checkboxes. Por exemplo, considere um formulário que coleta as linguagens de programação que os usuários conhecem. Para cada linguagem, podemos definir um checkbox. Para agrupar esses checkboxes, basta definir o atributo **name** com o mesmo valor para eles.

2. HTML – FORMULÁRIOS – CHECKBOXES E RADIOS



```
1 <input id="java_id" type="checkbox" value="java" name="linguagens">  
2 <input id="csharp_id" type="checkbox" value="csharp" name="linguagens">  
3 <input id="php_id" type="checkbox" value="php" name="linguagens">  
4 <input id="ruby_id" type="checkbox" value="ruby" name="linguagens">  
5 <input id="perl_id" type="checkbox" value="perl" name="linguagens">
```

☒ Java ☒ C# ☐ PHP ☒ Ruby ☐ Perl

2. HTML – FORMULÁRIOS – CHECKBOXES E RADIOS



```
1 <input id="sp_id" type="radio" value="sao-paulo" name="time-preferido">  
2 <input id="barcelona_id" type="radio" value="barcelona" name="time-preferido">  
3 <input id="milan_id" type="radio" value="milan" name="time-preferido">  
4 <input id="mu_id" type="radio" value="manchester-united" name="time-preferido">  
5 <input id="bdm_id" type="radio" value="bayern-de-munique" name="time-preferido">
```

☒ São Paulo ☐ Barcelona ☐ Milan ☐ Manchester United ☐ Bayern de Munique

Algumas vezes, desejamos que determinados checkboxes e radios estejam marcados quando os formulários são apresentados aos usuários. Para resolver esse problema, podemos utilizar o atributo **checked** do elemento **input**. Esse atributo não precisa de valor.

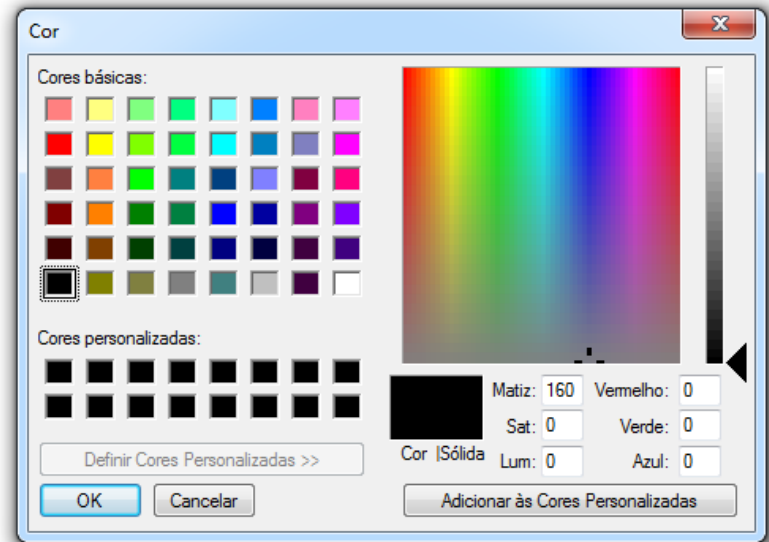
2. HTML – FORMULÁRIOS – SELEÇÃO DE CORES



No HTML5, para coletar uma cor, podemos utilizar o elemento **input** com **type** igual a **color**.



```
1 <input id="cor_id" name="cor" type="color">
```



2. HTML – FORMULÁRIOS – MENU DROP-DOWN



Muitos formulários permitem que os usuários selecionem um ou mais itens de uma lista de opções. Essa seleção pode ser realizada através de um **drop-down list**. Para adicionar esse tipo de componente, devemos utilizar o elemento **select**.

```
1 <select id="estados_id" name="estado">
2   ...
3 </select>
```


2. HTML – FORMULÁRIOS – MENU DROP-DOWN



As opções devem ser definidas no conteúdo do elemento **select** e elas são adicionadas com o elemento **option**. O conteúdo do elemento **option** é exibido para os usuários. Esse elemento possui um atributo chamado **value**. Quando o formulário for enviado, o valor do atributo **value** é transmitido ao Web Server se a opção correspondente foi selecionada pelo usuário. Se esse atributo não estiver definido, o conteúdo do elemento **option** é transmitido ao Web Server se a opção correspondente foi selecionada pelo usuário.

2. HTML – FORMULÁRIOS – MENU DROP-DOWN



```
1 <select id="estados_id" name="estado">
2   <option value="SP">São Paulo</option>
3   <option value="RJ">Rio de Janeiro</option>
4   <option value="RS">Rio Grande do Sul</option>
5   <option value="RN">Rio Grande do Norte</option>
6 </select>
```



São Paulo

- ✓ São Paulo
- Rio de Janeiro
- Rio Grande do Sul
- Rio Grande do Norte

2. HTML – FORMULÁRIOS – MENU DROP-DOWN



```
1 <select id="estados_id" name="estado" multiple="multiple">
2   <option value="SP">São Paulo</option>
3   <option value="RJ">Rio de Janeiro</option>
4   <option value="RS">Rio Grande do Sul</option>
5   <option value="PR">Paraná</option>
6   <option value="RN">Rio Grande do Norte</option>
7   <option value="BA">Bahia</option>
8 </select>
```

São Paulo

Rio de Janeiro

Rio Grande do Sul

Paraná

2. HTML – FORMULÁRIOS – MENU DROP-DOWN



```
1 <select id="estados_id" name="estado">
2   <optgroup label="Região Sudeste">
3     <option value="SP">São Paulo</option>
4     <option value="RJ">Rio de Janeiro</option>
5   </optgroup>
6   <optgroup label="Região Sul">
7     <option value="RS">Rio Grande do Sul</option>
8     <option value="PR">Paraná</option>
9   </optgroup>
10  <optgroup label="Região Nordeste">
11    <option value="RN">Rio Grande do Norte</option>
12    <option value="BA">Bahia</option>
13  </optgroup>
14 </select>
```



2. HTML – FORMULÁRIOS – FIELDSET



Os campos de um formulário muito longo podem ser agrupados logicamente com o elemento **fieldset**. No conteúdo desse elemento, podemos utilizar o elemento **legend** para definir a legenda do fieldset.

```
1 <fieldset>
2   <legend>Dados Pessoais</legend>
3   ...
4 </fieldset>
5 <fieldset>
6   <legend>Formação</legend>
7   ...
8 </fieldset>
9 <fieldset>
10  <legend>Experiência</legend>
11  ...
12 </fieldset>
```

2. HTML – FORMULÁRIOS – FIELDSET



Dados Pessoais

...

Formação

...

Experiência

...

2. HTML – FORMULÁRIOS – AUTOCOMPLETE



Para melhorar a usabilidade, podemos utilizar o recurso do autocomplete nas caixas de entrada. Para utilizar esse recurso, devemos criar uma lista de sugestões com o elemento **datalist**. É fundamental identificar as listas de sugestões com o atributo **id**.

As opções devem ser definidas no conteúdo do elemento **datalist** e elas são adicionadas com o elemento **option**. O atributo **value** de um elemento **option** define uma sugestão.

Com a lista de sugestões criada, podemos associá-la a uma caixa de entrada através do atributo **list** do elemento **input**.

2. HTML – FORMULÁRIOS – AUTOCOMPLETE



```
1 <datalist id="comidas_id">
2   <option value="Lasanha">
3   <option value="Pizza">
4   <option value="Sopa">
5   <option value="Salada">
6 </datalist>
```

Comida: s

Sopa

Salada

```
1 <input id="comida_id" name="comida" type="text" list="comidas_id">
2
3 <datalist id="comidas_id">
4   <option value="Lasanha">
5   <option value="Pizza">
6   <option value="Sopa">
7   <option value="Salada">
8 </datalist>
```


2. HTML – FORMULÁRIOS – VALIDAÇÃO



Alguns recursos para realizar a validação dos campos de um formulário foram adicionados no HTML5. Essas validações ocorrem antes do envio dos formulários. Por padrão, algumas validações são realizadas automaticamente de acordo como tipo do campo. Por exemplo, os navegadores verificam se o conteúdo de uma caixa de email é um email válido.

```
1 <input id="email_id" name="email" type="email">
```

A screenshot of a web form with the label 'E-mail:' followed by a text input field containing 'xpto'. To the right of the input field is a button labeled 'Enviar'. Below the input field, a white tooltip box with a yellow exclamation mark icon contains the text 'Please enter an email address.'.

E-mail:

! Please enter an email address.

2. HTML – FORMULÁRIOS – VALIDAÇÃO



Os navegadores também verificam se o conteúdo de uma caixa de URL é uma url válida

```
1 <input id="url_id" name="url" type="url">
```

2. HTML – FORMULÁRIOS – VALIDAÇÃO



Outras validações devem ser definidas explicitamente. Por exemplo, se um determinado campo é obrigatório, devemos utilizar o atributo **required**. Esse atributo não precisa de valor.

```
1 <input id="nome_id" name="nome" type="text" required>
```

A screenshot of a web form. It contains a label 'Nome:' followed by a text input field. To the right of the input field is a button labeled 'Enviar'. Below the input field, a validation error message is displayed in a white box with a yellow border. The message starts with an orange exclamation mark icon and says 'Please fill out this field.'

2. HTML – FORMULÁRIOS – VALIDAÇÃO



Se um determinado campo deve respeitar uma expressão regular, devemos utilizar o atributo **pattern**.

```
1 <input
2   id="placa_id"
3   name="placa"
4   type="text"
5   pattern="[A-Z]{3} [0-9]{4}"
6   title="As placas são formadas por 3 letras ou 4 números">
```

Placa:

! Please match the requested format.
As placas são formadas por 3 letras ou 4 números

2. HTML – FORMULÁRIOS – VALIDAÇÃO



Se o conteúdo de um campo numérico deve estar em um determinado intervalo, podemos utilizar os atributos **min** e **max**.

```
1 <input
2   id="idade_id"
3   name="idade"
4   type="number"
5   min="18"
6   max="60"
7   title="A idade mínima é 18 e a máxima é 60">
```

Idade:



Value must be greater than or equal to 18.

A idade mínima é 18 e a máxima é 60

2. HTML – FORMULÁRIOS – VALIDAÇÃO



Se o conteúdo de um campo numérico deve estar em um determinado intervalo, podemos utilizar os atributos **min** e **max**.

```
1 <input
2   id="idade_id"
3   name="idade"
4   type="number"
5   min="18"
6   max="60"
7   title="A idade mínima é 18 e a máxima é 60">
```

Idade:



Value must be greater than or equal to 18.

A idade mínima é 18 e a máxima é 60

2. HTML – DIVISÕES LÓGICAS E ESTRUTURAIS



O nome **div** vem de divisão, e esta tag vai lhe dar o simples poder de **dividir qualquer trecho de seu código**. Isso mesmo, você pode criar um bloco, uma divisão, e dentro deste bloco ter uma imagem, links, textos e o que mais você quiser.

E qual a vantagem disso? Você vai poder aplicar formatação **CSS** nesse bloco, e tudo que estiver dentro da **div**, seja imagem, texto, link ou o que mais for, vai receber aquelas regras de estilo.

As **divs** são **IMPORTANTÍSSIMAS** para estruturar, criar uma ordem lógica e organizar um site.

Você cria uma div para o cabeçalho, outra para o conteúdo, outra div para os menus e uma última para o rodapé, então usa o CSS para estilizar cada uma dessas divs separadamente.

2. HTML – DIVISÕES LÓGICAS E ESTRUTURAIS



A sintaxe da tag div é:

<div>

Todo, qualquer e quantos
elementos do HTML você queira
colocar aqui.

</div>

2. HTML – DIVISÕES LÓGICAS E ESTRUTURAIS



A sintaxe da *span* tag é como outra tag qualquer, a diferença é que devemos envolver o texto que quisermos entre: `` e ``.

A utilização desta tag é mais usual com os estilos CSS que será estudado no próximo no decorrer do curso.

```
<p>Para que serve a tag span - Tutorial de <span style="color:blue">HTML</span> e <span style="color:red">CSS</span></p>
```

E o efeito ocorrido nas tags *span* estilizadas será:

Para que serve a tag span - Tutorial de HTML e CSS

OBRIGADO !!!
PERGUNTAS ?