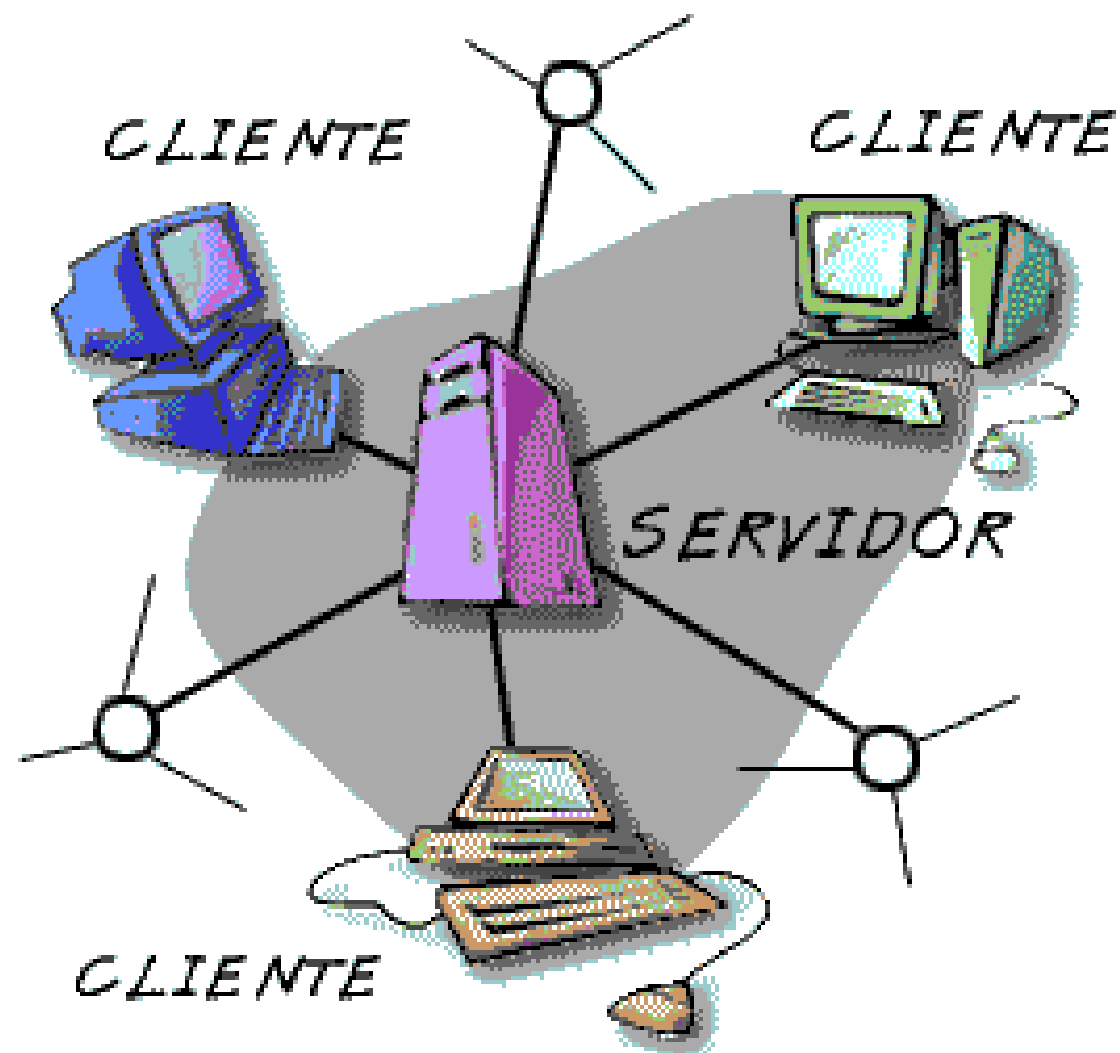




Disciplina - HTML, CSS e Javascript



Arquitetura



Vantagens

- Portabilidade da solução no lado do cliente
- Facilidade de *deployment*
- Facilidade de manutenção, restauração e atualização da aplicação

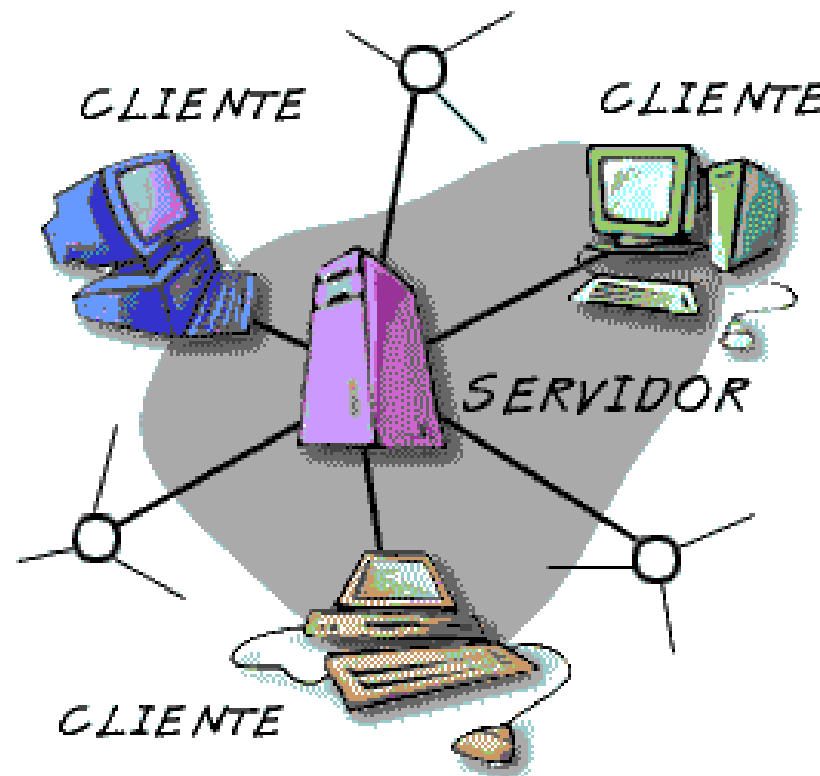


Desvantagens

- Número expressivo e crescente de dispositivos diferentes com acesso a web (computação ubíqua)
- Compatibilidade entre browsers
- Novos desafios para a Engenharia de Software:
 - Adequação ao “internet time”
 - Metodologias voltadas para o desenvolvimento web
 - Computação Concorrente

Execução no Cliente (Browser)

- HTML
- CSS
- Tableless
- JavaScript
- XML ou JSON



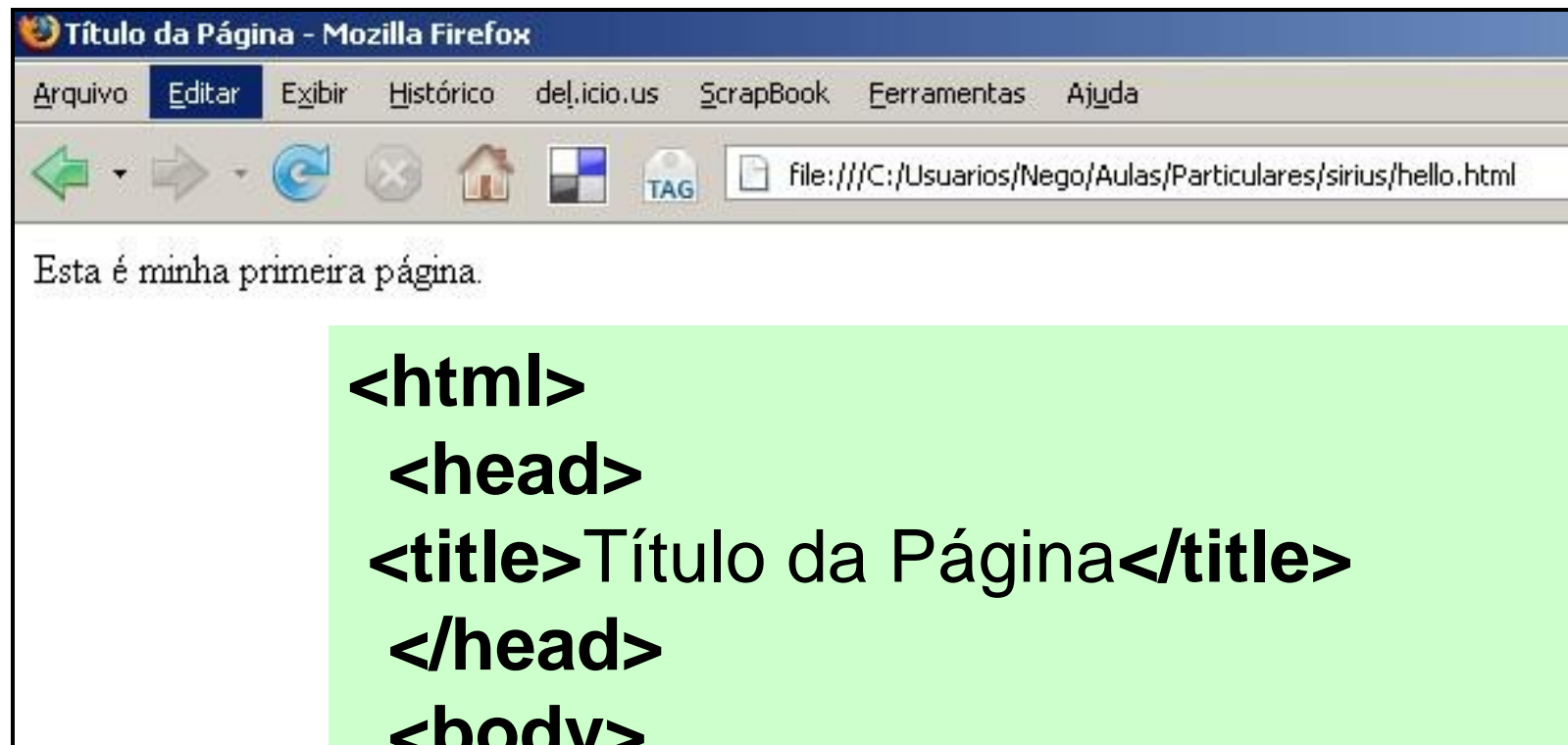


HTML

- *Hyper Text Markup Language*
- Especificação definida pelo consórcio W3C:
<http://www.w3.org/>
- Um arquivo html contém marcadores (tags)
- Estes marcadores indicam para o navegador (browser) como a página deve ser apresentada
- Marcadores usualmente vem em pares:
`<tag>...</tag>`
- Também podem aparecer de forma abreviada: `<tag atributo="valor" ... />`



HTML



```
<html>
  <head>
    <title>Título da Página</title>
  </head>
  <body>
    Esta é minha primeira página.
  </body>
</html>
```



HTML

Tags Básicas

Tag	Descrição
<u><html></u>	Define um documento HTML
<u><body></u>	Define o corpo de um documento
<u><h1> to <h6></u>	Define cabeçalhos 1 a 6
<u><p></u>	Define um parágrafo
<u>
</u>	Insere uma quebra de linha
<u><hr></u>	Define uma linha horizontal
<u><!-- --></u>	Define um comentário



HTML

Tags de Formatação

Tag	Descrição
<u></u>	Formata um texto em negrito
<u><big></u>	Formata um texto com fonte maior
<u></u>	Formata um texto com <i>ênfase</i>
<u><i></u>	Formata um texto em <i>itálico</i>
<u><small></u>	Formata um texto com fonte pequena
<u></u>	Formata um texto em destaque
<u><sub></u>	Formata um texto _{subscrito}
<u><sup></u>	Formata um texto ^{sobrescrito}
<u><ins></u>	Formata um texto <u>sublinhado</u>
<u></u>	Formata um texto anulado

HTML

Entidades

- Utilizadas para apresentação de caracteres especiais em html. Ex.: “<”

Saída	Descrição	Nome da Entidade	Número
	Espaço sem quebra	 	
<	Menor que	<	<
>	Maior que	>	>
&	E comercial	&	&
"	Aspas	"	"
'	Apóstrofo	' (does not work in IE)	'

HTML

Links e Imagens

- Link -

`Página da Sirius!`

– [Página da Sirius](http://www.sirius.com.br)

- Ancora

`Como Chegar`

– Trecho da página que informa detalhes sobre localização

- ``

`Localização`

– [Localização](#)

- ``

HTML

Tabelas

```
<table border="1">  
<tr>  
  <th>Coluna 1</th>  
  <th>Coluna 2</th>  
</tr>  
<tr>  
  <td>linha 1, valor 1</td>  
  <td>linha 1, valor 2</td>  
</tr>  
<tr>  
  <td>linha 2, valor 1</td>  
  <td>linha 2, valor 2</td>  
</tr>  
</table>
```

Coluna 1	Coluna 2
linha 1, valor 1	linha 1, valor 2
linha 2, valor 1	linha 2, valor 2

HTML

Background e Fontes

- Formas de se definir um background preto para o corpo da página
 - `<body bgcolor="#000000">`
 - `<body bgcolor="rgb(0,0,0)">`
 - `<body bgcolor="black">`
- Definindo uma imagem de fundo
 - `<body background="logo.gif">`
 - `<body
background="http://www.sirius.com.br/clouds.gif">`
- Configurando uma fonte
 - `Texto com fonte específica.`

HTML

Formulários

- HTML possibilita a criação de formulários online utilizando tags
- A tag <form> é a mais comum e permite a criação de um formulário de entrada de dados

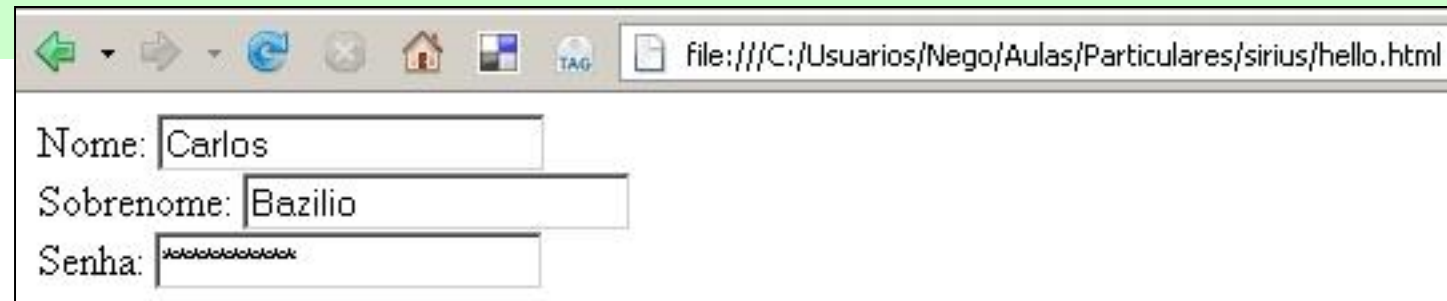
```
<body><form>
```

```
Nome: <input type="text" name="firstname"><br />
```

```
Sobrenome: <input type="text" name="lastname"><br />
```

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

```
</form></body>
```



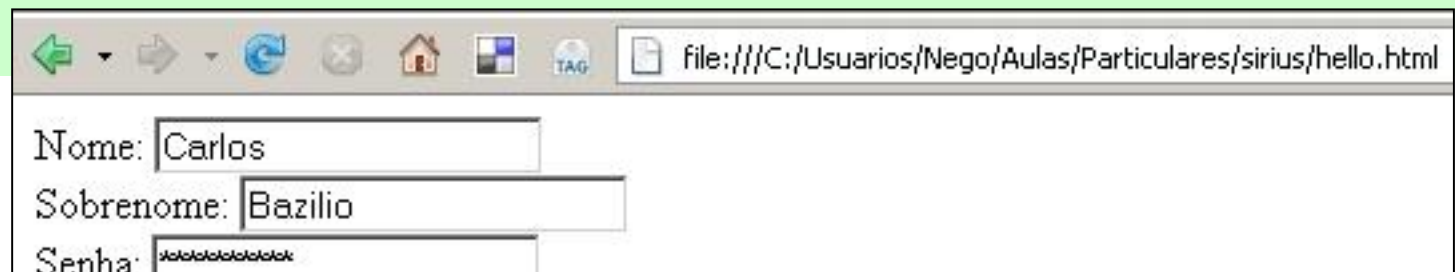
A screenshot of a web browser window showing a simple HTML form. The browser's address bar displays the file path: file:///C:/Usuarios/Nego/Aulas/Particulares/sirius/hello.html. The form contains three input fields: 'Nome' with the value 'Carlos', 'Sobrenome' with the value 'Bazilio', and 'Senha' which is masked with asterisks. The browser's toolbar includes back, forward, and refresh buttons, as well as icons for home, search, and a TAG extension.

HTML

Formulários

- O primeiro atributo é o **action** que serve para **definirmos o nome do arquivo que receberá os dados preenchidos no formulário**. O único objetivo desse atributo é informar para aonde serão enviados os dados do formulário e esse arquivo que receberá os dados, criado em linguagem java, por exemplo, ficará responsável por analisar, processar e/ou armazenar as informações.

```
<body><form action="gravar_dados">  
Nome: <input type="text" name="firstname"><br />  
Sobrenome: <input type="text" name="lastname"><br />  
Senha: <input type="password" name="senha">  
</form></body>
```



A screenshot of a web browser window showing the rendered HTML form. The browser's address bar displays the file path: `file:///C:/Usuarios/Nego/Aulas/Particulares/sirius/hello.html`. The form contains three input fields: a text field for "Nome" with the value "Carlos", a text field for "Sobrenome" with the value "Bazilio", and a password field for "Senha" with masked characters (dots).

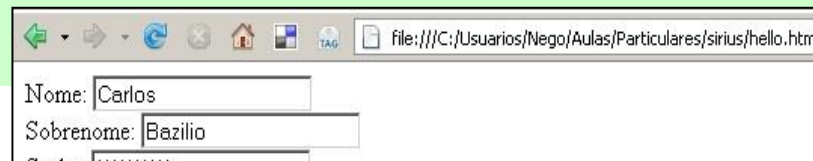


HTML

Formulários

- O segundo atributo também obrigatório em um formulário é o **method**, que serve para especificarmos a forma de envio dos dados, podendo ser do tipo **GET** ou **POST**. Segue abaixo a definição de ambos:
- **GET** – método que envia as variáveis digitadas pelo usuário pela **URL**, ou seja, podemos ver as variáveis sendo passadas pela URL da página de destino. Não é muito aconselhável o uso do método GET, pois ele expõe os nomes e os valores das variáveis.
- **POST** – método que envia as variáveis digitadas pelo corpo do protocolo, sendo completamente transparente para o usuário. É o método mais aconselhável.

```
<body><form method="post">  
Nome: <input type="text" name="firstname"><br />  
Sobrenome: <input type="text" name="lastname"><br />  
Senha: <input type="password" name="senha">  
</form></body>
```





HTML

Formulários

- Tipos que podem ser utilizados com a tag `<input>`:
 - button: Insere um botão
 - checkbox: Insere um checkbox; para vários itens, basta inserirmos vários “inputs” deste tipo
 - file: Insere botão seleção de arquivo através de uma caixa de diálogo
 - hidden: Campo pertencente ao formulário, mas que não será exibido na página
 - image: Insere uma imagem como um botão submit
 - password: Insere um campo password (caracteres digitados não aparecem)
 - radio: Insere um radiobox (análogo ao checkbox)
 - reset: Restaura os valores iniciais do formulário
 - submit: Encaminha os dados inseridos para algum arquivo
 - text: Insere um campo de edição de texto



HTML

Formulários

- Quando algum elemento do tipo “submit” é inserido num formulário e acionado, seus dados são enviados para um arquivo
- O arquivo mencionado é indicado pelo atributo “action” do elemento <form>
- Este arquivo deverá estar armazenado num servidor web (Apache, Tomcat, IIS, ...), e estará escrito em alguma linguagem de programação de servidores:
jsp, php, asp, sevlets, ...



HTML

Formulários

```
<body>
```

```
<form action="processaForm.jsp" >
```

```
Nome: <input type="text" name="firstname"><br />
```

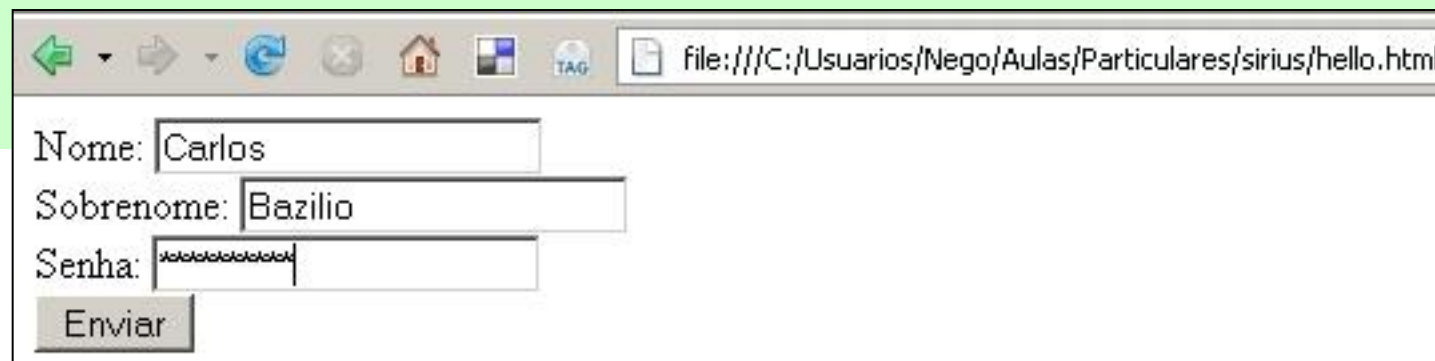
```
Sobrenome: <input type="text" name="lastname"><br />
```

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

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

```
</form>
```

```
</body>
```



A screenshot of a web browser window showing a form. The address bar displays the file path: file:///C:/Usuarios/Nego/Aulas/Particulares/sirius/hello.html. The form contains four input fields: 'Nome:' with the value 'Carlos', 'Sobrenome:' with the value 'Bazilio', and 'Senha:' with a masked password 'xxxxxxxxxx'. Below the fields is a button labeled 'Enviar'.

- URL após clicar no botão:
.../sirius/processaForm.jsp?firstname=Carlos&lastname=Bazilio&senha=abcdefg



Caixas de números

- › Para coletar dados numéricos, podemos utilizar caixas específicas para números.
- › Valor do atributo `type` é **number** ou **range**.

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

Idade: Quantidade:



Caixas de email, telefone 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 e-mails, telefones e urls respectivamente.

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



Caixas de email, telefone e url

192.168.1.118/~keizo/html/pub ↻ 📄 ⋮

E-mail:

Telefone:

Site:

Enviar

q w e r t y u i o p
a s d f g h j k l
⬆ z x c v b n m ⬆
?123 @ . Ir

192.168.1.118/~keizo/html/pub ↻ 📄 ⋮

E-mail:

Telefone:

Site:

Enviar

1 2 ABC 3 DEF -
4 GHI 5 JKL 6 MNO .
7 PQRS 8 TUV 9 WXYZ ⬆
* # 0 + _ Avançar

192.168.1.118/~keizo/html/pub ↻ 📄 ⋮

E-mail:

Telefone:

Site:

Enviar

q w e r t y u i o p
a s d f g h j k l
⬆ z x c v b n m ⬆
?123 🎤 . Ir



Caixas de datas e horas

Diversos tipos de caixas de entrada para coletar datas e horas foram adicionados no HTML5:

- › `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.
- › `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.



Caixas de datas e horas

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




Caixas de datas e horas

mm/dd/yyyy : ▾ Enviar

September 2013 ▾ ◀ ● ▶

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

mm/dd/yyyy --:--:-- : ▾ Enviar

September 2013 ▾ ◀ ● ▶

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

Figure:date e datetime-local

Caixas de datas e horas

Week

September 2013

Week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
36	1	2	3	4	5	6	7
37	8	9	10	11	12	13	14
38	15	16	17	18	19	20	21
39	22	23	24	25	26	27	28
40	29	30	1	2	3	4	5

September 2013 ▾

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

Figure:week e month

Caixas de datas e horas

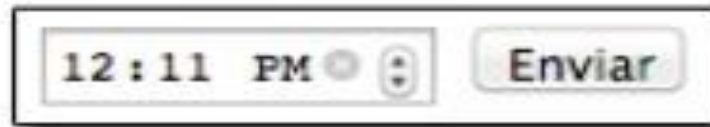


Figure:time



Caixas de senha

- › Para adicionar uma caixa de senha em um formulário, devemos utilizar o elemento `input` com o valor `password` para o atributo `type`.

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

Senha:



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`.

```
1 <textarea id="mensagem_id" name="mensagem" maxlength="140">  
2  
3 </textarea>
```



Checkboxes

- › Para adicionar um checkbox em um formulário, devemos utilizar o elemento input com type igual a checkbox.

```
1 <input id="sim_id" name="newsletter" type="checkbox" value="sim">
```



Checkboxes

- › Para agrupar esses checkboxes, basta definir o atributo name com o mesmo valor para eles.

```
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

Radios

- › Para adicionar um radio em um formulário, devemos utilizar o elemento input com type igual a radio.
- › Ao utilizar esse componente, é importante definir um valor para o atributo value.
- › Algumas vezes, desejamos que determinados checkboxes e radios estejam marcados quando os formulários são apresentados aos usuários: checked

```
<form>  
  <input type="radio" name="sex" value="male" checked>Male  
  <br>  
  <input type="radio" name="sex" value="female">Female  
</form>
```

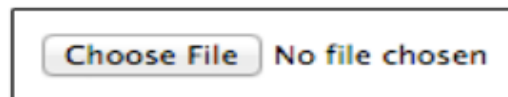
☒ Male
☐ Female



Botoões de upload

- › Para adicionar um botão de upload em um formulário, podemos utilizar o elemento input com type igual a file.

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





Drop-down list

- › Muitos formulários permitem que os usuários selecionem um ou mais itens de uma lista de opções.
- › Para adicionar esse tipo de componente, devemos utilizar o elemento `select`.
- › As opções devem ser definidas no conteúdo do elemento `select` e elas são adicionadas com o elemento `option`.



Drop-down list

```
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>
```



Drop-down list





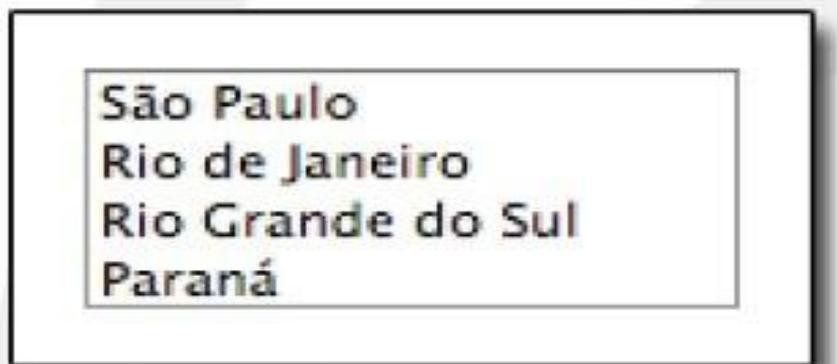
Drop-down list

- › Por padrão, apenas um item de um drop-down list pode ser selecionado pelos usuários. Mas, utilizando o atributo `multiple`, um ou mais itens podem ser selecionados.

```
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>
```



Drop-down list



São Paulo
Rio de Janeiro
Rio Grande do Sul
Paraná



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`.



Autocomplete

```
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>
```

Comida:

- Sopa
- Salada



Validação

- › Validação: verificar se os dados fornecidos no formulário correspondem minimamente com que é solicitado ao usuário
- › Segurança, organização de banco de dados ou até para evitar a perda de tempo na hora de filtrar entre dezenas de mensagens spam.
- › Tarefa trabalhosa que demanda tempo.
- › HTML5 a validação tornou-se nativa da linguagem



Validação

- › Alguns recursos para realizar a validação dos campos de um formulário foram adicionados no HTML5.

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

E-mail: xpto

Enviar



Please enter an email address.



Validação

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

URL:

 Please enter a URL.



Validação

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

Nome:

Enviar



Please fill out this field.



HTML 5

Tag	Descrição
<article>	Especifica um artigo qualquer
<aside>	Relaciona um conteúdo ao redor.
<bdi>	Para texto que não esteja diretamente vinculado ao elemento pai.
<command>	Um botão, radioButton ou checkBox.
<details>	
<summary>	Uma legenda ou resumo dentro do elemento detalhe.
<figure>	Para agrupamento de uma sessão de conteúdo stand-alone.
<figcaption>	Legenda para um elemento figura.
<footer>	Para um rodapé de um documento ou sessão.
<header>	Para uma introdução de um documento ou sessão.
<hgroup>	Para uma sessão de cabeçalhos usando h1 para h6.
<mark>	Para textos que são destaques.
<meter>	Para uma medição. Deve-se conhecer o valor máximo e mínimo./td>
<nav>	Para uma sessão de navegação./td>
<progress>	Para mostrar o progresso de uma execução./td>
<ruby>	Para anotação rubi./td>
<rt>	Para explicar uma anotação rubi./td>
<rp>	Mostra os navegadores que não suportam anotação rubi./td>
<section>	Para uma sessão de um documento como capítulos, cabeçalhos, rodapés, etc./td>
<time>	Para definir um tempo e/ou data.
<wbr>	Quebra de linha.

HTML 5 - Formulário

Type	Descrição
<code>tel</code>	A entrada é um número de telefone.
<code>search</code>	A entrada é um campo de busca.
<code>url</code>	A entrada é uma URL.
<code>email</code>	A entrada é um ou mais endereço de email.
<code>datetime</code>	A entrada é uma data e/ou hora.
<code>date</code>	A entrada é uma data
<code>month</code>	A entrada é um mês.
<code>week</code>	A entrada é um dia da semana.
<code>time</code>	A entrada é uma hora.
<code>datetime-local</code>	A entrada é uma data e hora local.
<code>number</code>	A entrada é um número.
<code>range</code>	A entrada é uma faixa de valores.
<code>color</code>	A entrada é uma cor em hexadecimal como #FF00FF.
<code>placeholder</code>	Especifica uma dica rápida que descreve o valor esperado em um campo de entrada.



HTML

Outras tags de Formulários

Tag	Descrição
<form>	Define um formulário para entrada do usuário
<input>	Define um campo de entrada
<textarea>	Define um campo texto com múltiplas linhas
<label>	Define um label para algum controle
<fieldset>	Desenha uma caixa em torno de um conjunto de elementos
<legend>	Define um título para um <fieldset>
<select>	Cria uma lista drop-down
<optgroup>	Criar uma hierarquia nas opções de uma lista drop-down
<option>	Uma opção na lista drop-down
<button>	Insere um botão. Difere de <input> por poder conter algum conteúdo, como uma imagem

HTML

Outras tags de Formulários

Netscape - [Get Me Rich Quick]

File Edit View Go Bookmarks Options Directory Window Help

Get Me Rich Quick

Check this checkbox if you want to send me money: ☒

Enter your credit card number in this text box:

Expiration date:

Select the card type with these radio buttons:

☒ Visa
☐ MasterCard
☐ Discover

Choose the amount of your contribution from
this scrolling list:
\$1,000,000
\$100,000
\$10,000

Pick your reason from this pull-down list:

Enter any comments in this text area:

Gosh, this is a rare privilege.
Thanks and enjoy the dough.

Document Done



+Fontes

A página HTML é executada no computador do usuário, a apresentação depende da configuração do sistema operacional e do navegador utilizado.

Por precaução use mais de uma fonte:

```
<font face="verdana, arial, helvetica" size=2 color=000000>
```

texto em verdana ou arial ou helvetica tamanho 2, cor preta

```
</font>
```

O formato padrão "Times New Roman".

Destaque

- ❖ `<i>` itálico `</i>`
- ❖ `` negrito ``
- ❖ `` negrito ``
- ❖ `<big>` grande `</big>`
- ❖ `<small>` pequeno `</small>`
- ❖ `<u>` underline `</u>`
- ❖ `^{` superescrito `}`
- ❖ `_{` subscrito `}`
- ❖ `<strike>`taxado`</strike>`
- ❖ `<tt>`mono-espaçado`
</tt>`
- ❖ ``ênfase``
- ❖ ``ênfase forte``



Outros destaques

<marquee> rolagem </marquee>

<hr size=8 width=400>

<hr size="5" width="50%"> espaço

<p> parágrafo </p>

<h1>titulo1</h1>

<h6>titulo 6</h6>



Marcador

```
<ol>
```

```
<li> ítem 1 </li>
```

```
<ul type="circle">
```

```
  <li> subítem 1</li>
```

```
  <ul>
```

```
    <li> subsubítem</li>
```

```
  </ul>
```

```
  <li> subítem 2</li>
```

```
</ul>
```

```
<li> ítem 2 </li>
```

```
<ol style="margin-top:0cm;" start="1" type="a">
```

```
  <li> subítem </li>
```

```
</ol>
```

```
<li> ítem 3 </li>
```

```
</ol>
```

IMAGEM

A ``tag é usada para incorporar uma imagem em uma página HTML.

``

src - Especifica o caminho para a imagem



Lista de exercícios



CSS



CSS

- *Cascading Style Sheets*
- Estilos definem como elementos html devem ser apresentados
- Permite a separação entre definição de conteúdo e formatação em HTML
- Style sheets externos são definidos através de arquivos CSS
- Atualmente é o padrão para inserção de estilo na construção de páginas html
- <http://www.csszengarden.com/>

CSS

- Formato geral:
`seletor { propriedade: valor }`
- Uma definição de estilo em CSS será composta por uma sequência de definições como esta acima
- Exemplos:
 - `body { color: black }`
 - `p { font-family: "Verdana"; text-align: center; color: red }`
 - `h1,h2,h3,h4,h5,h6 { color: green }`
 - `p {margin-left: 20px}`

CSS

Como definir (1/3)

- Existem 3 maneiras se definir um CSS para um documento HTML
 - Criando um link externo

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="mystyle.css" />
  </head>
  <body>
    Esta é minha primeira página.
  </body>
</html>
```

```
/* Conteúdo do arquivo mystyle.css */
hr {color: sienna}
p {margin-left: 20px}
body {background-image:
  url("images/back40.gif")}
```

CSS

Como definir (2/3)

- Internamente (dentro da própria página HTML a ser exibida)

```
<html>
  <head>
    <style type="text/css">
      hr {color: sienna}
      p {margin-left: 20px}
      body {background-image: url("images/back40.gif")}
    </style>
  </head>
  <body>
    Esta é minha primeira página.
  </body>
</html>
```

CSS

Como definir (3/3)

- Definindo de forma inline

```
<html>  
  <head></head>  
  <body style="color: sienna; text-align: center;">  
    Esta é minha primeira página.  
  </body>  
</html>
```

- Caso existam várias definições para um mesmo elemento, a prioridade é: inline, interna e externa



Aplicações

- Com CSS podemos formatar:
 - Background
 - Textos
 - Fontes
 - Margens
 - Bordas
 - Listas
 - Tabelas

http://www.w3schools.com/css/css_examples.asp

Uma regra pode ter mais que uma declaração

```
1  
2 p {  
3  
4     font-size: 14px; /* a fonte do texto tem 14 pixels de tamanho */  
5  
6     color: red; /* a cor da fonte é vermelha */  
7  
8 }  
9
```



CSS

- Classes em CSS permitem que um mesmo elemento seja exibido de diferentes formas
- Formato usando classes:
`seletor.classe { propriedade: valor }`
- Exemplos:
 - `p.direita {text-align: right}` // Alinha à direita
 - `p.centro {text-align: center}` // Centraliza
 - `.esquerda {text-align: left}` // Aplicado a qualquer elemento html que contenha classe *esquerda*

```
<p class="direita">Este parágrafo será alinhado à direita</p>
```

```
<p class="centro">Este parágrafo será centralizado</p>
```

```
<p class="esquerda">Este será à esquerda</p>
```

Uma regra pode ter mais de um seletor

```
1  
2 /* Os parágrafos e cabeçalhos h1, h2, h3 possuem cor da fonte  
3 vermelha */  
4  
5 p , h1, h2, h3 {  
6  
7     color: red;  
8  
9 }
```


Uma regra pode ter mais de um seletor

body {

/* aqui vai uma declaração qualquer */

}

div {

/* aqui vai uma declaração qualquer */

}

p, span, strong {

/* aqui vai uma declaração qualquer */

}



Seletor tipo ID

```
#nome-do-identificador {  
  
    background-color: green; /* cor de fundo verde */  
  
}
```

```
1  
2 <body id="nome-do-identificador">  
3 </body>  
4
```



Seletor tipo ID

```
.nome-da-classe {  
  
    color: blue; /* cor de texto azul */  
  
}
```

1	<h1 class="nome-da-classe">Título com cor azul</h1>
2	
3	<p class="nome-da-classe">Parágrafo com a cor azul.</p>
4	



Seletor de atributo

```
input[type='submit'] {  
  
    font-weight: bold; /* texto em negrito */  
  
}
```

```
1 <input type="submit" value="Enviar">  
2  
3  
4
```



Margin, border, padding, content

```
p {  
  width: 220px;  
  padding: 10px;  
  border: 5px solid  
  gray;  
  margin: 20px;  
}
```



Medidas de Comprimento

- ❑ **Medidas relativas** – variam conforme a resolução da tela
 - ❑ *Pixels (px)* – dependente do dispositivo
 - ❑ Relativas ao tamanho da fonte:
 - ❑ *em* (mesmo tamanho que o tamanho da fonte para o elemento em que ela está inserida)
 - ❑ *ex* (altura da letra "x" minúscula para a fonte do elemento em que ela está inserida)
 - ❑ *Porcentagem (%)*

- ❑ **Medidas absolutas** – variação de tamanho não é baseada nas propriedades do sistema (não é herdada, não é relativa a outra medida)
 - ❑ Polegadas (*in*)
 - ❑ Centímetros (*cm*)
 - ❑ Milímetros (*mm*)
 - ❑ Pontos (*pt*) – $1\text{ pt} = 1/72\text{ in}$
 - ❑ Paicas (*pc*) – $1\text{ pc} = 12\text{ pt}$



Tableless

- **Padrão que vem sendo adotado na web para uso massivo de html+css**
- Objetiva, principalmente, melhor acessibilidade de páginas web
- **Propõe a não utilização tabelas html para a construção do layout da página** (origem do nome)
- Naturalmente, este uso não deveria ser evitado se o que se deseja construir é uma tabela de fato



Objetivo: montar um layout assim



ABOUT

SERVICES

PORTFOLIO

CONTACT US

enlightendesigns

ABOUT

Enlighten Designs is an Internet solutions provider that specialises in front and back end development. To view some of the web sites we have created view our [portfolio](#).

We are currently undergoing a 'face lift', so if you have any questions or would like more information about the services we provide please feel free to contact us.

CONTACT US

Phone: (07) 853 6060
Fax: (07) 853 6060
Email: info@enlighten.co.nz
P.O Box: 14159, Hamilton, New Zealand

[More contact information...](#)

FEATURED PROJECTS

The New Zealand National Party

We recently launched the [New Zealand National Party](#) website - Lorem ipsum dolor sit amet. Pita ala pesama borak na falanuisen.

Greensets

We recently launched the [Greensets](#) website - Lorem ipsum dolor sit amet. Pita ala pesama borak na falanuisen.

Nutrition Care

We recently launched the [Nutrition Care](#) website - Lorem ipsum dolor sit amet. Pita ala pesama borak na falanuisen.

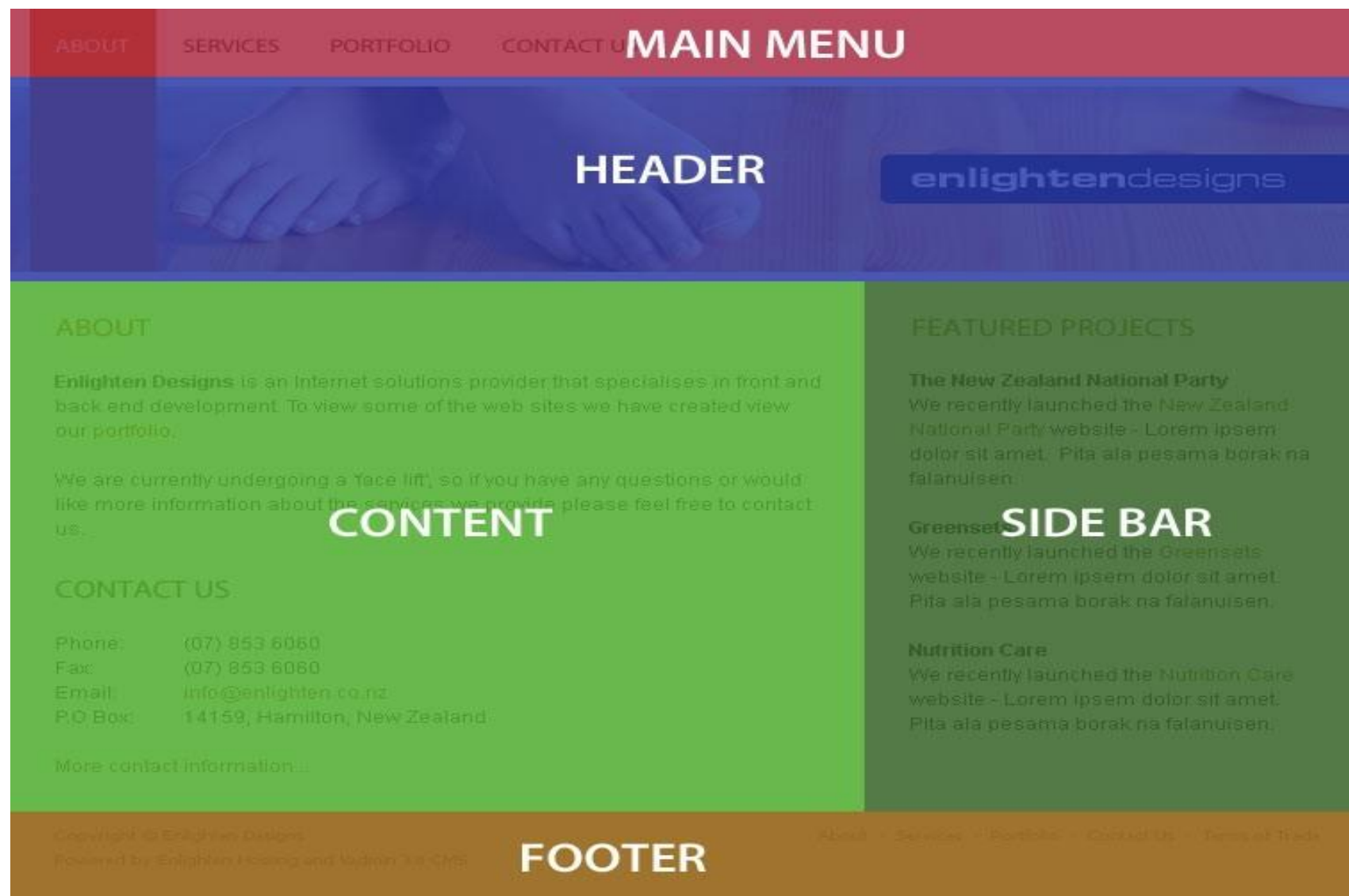


Identificar os elementos estruturais

- Antes de tudo, temos que identificar os elementos estruturais do design, para então saber como estruturar nosso documento HTML.
- A web é amplamente baseada em retângulos, e nós precisamos nos lembrar disso quando estivermos dividindo nosso design. Essas divisões principais que nós iremos fazer vão terminar em tags `<div>`. Um `<div>` é basicamente um contêiner retangular, o qual podemos posicioná-lo usando CSS.



Dividindo o design





5 elementos principais:

- **nav** (área de navegação principal):
A navegação primária para este website.
As imagens vão mudar on hover (quando o cursor do mouse estiver sobre elas).
Comprimento: 760px Altura: 50px
- **header** (cabeçalho): O cabeçalho do website inclui uma imagem de fundo (puramente por estética) e o nome da companhia.
Comprimento: 760px Altura: 150px
- **footer** (rodapé): Informações gerais: Copyright, créditos, e algum texto alternativo para navegação.

Comprimento: 760px Altura: 66px
- **article** (conteúdo): É nesta área que ficará o conteúdo principal do website.
Comprimento: 480px
Altura: Pode mudar, dependendo do conteúdo
- **aside** (barra lateral): Esta terá uma camada secundária que não é tão importante quanto o conteúdo principal.
Comprimento: 280px Altura: Pode mudar, dependendo do conteúdo

Este site será também centralizado na janela do navegador.



index.html

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" type="text/css" href="style.css">
    <title>Meu Site</title>
  </head>
  <body>
    <nav>Main Nav</nav>
    <header>Header</header>
    <aside>
      Texto texto texto texto texto texto texto texto texto texto texto texto texto texto
      texto texto texto texto texto texto texto.
    </aside>
    <article>
      Texto texto texto texto texto texto texto texto texto texto texto texto texto texto
      texto texto texto texto texto texto texto texto texto texto texto texto texto texto
      texto texto texto texto texto texto texto texto texto texto.
    </article>
    <footer>Footer</footer>
  </body>
</html>
```



- Como você vai notar no design, qualquer coisa em nossa página é da largura de 760px ou menos, e nada flutua fora disso. O que nós vamos fazer é criar um contêiner para nossa página que tenha 760px de largura e posicionado no meio da página. Nossos 5 principais elementos serão colocados dentro desse contêiner.



style.css

- E isto é todo o HTML que precisamos para nosso contêiner. Vamos à CSS.
- Crie um novo arquivo vazio e salve-o como estilo.css.



ID e CLASS

- O sinal **#** na frente da id diz ao navegador que estamos selecionando uma **id**. Para uma **classe**, nós usaríamos um **.** Exemplo: `.page-container { }` — no caso de uma classe.
- Um **id** é um **identificador único** que usamos para coisas que somente ocorrerão uma vez na página. Então para cabeçalhos, rodapés, navegadores, etc..., nós usamos id's e para elementos que se repetem, como links, nós devemos usar classes, que podem aparecer mais vezes na mesma página.
Nós não estaremos habilitados a ver as modificações
- que são feitas a esta **<div>**, porque ela é transparente por padrão.



- Precisamos escrever uma regra CSS para redefinir as margens e alinhamentos das tags `html` e `body` para zero. Adicione essa regra bem no topo do seu arquivo `css`:

```
html, body {  
    margin: 0;  
    padding: 0;  
}
```




Os CINCO elementos principais



- Nós precisamos adicionar 5 tags.
- Estas tags correspondem às áreas do design que nós identificamos na parte 2.
- Nós vamos inserir texto dentro das tags somente para facilitar a identificação visual quando visualizarmos a página.



HTML

```
<div id="page-container">  
  <div id="main-nav">Main Nav</div>  
  <div id="header">Header</div>  
  <div id="sidebar-a">Sidebar A</div>  
  <div id="content">Content</div>  
  <div id="footer">Footer</div>  
</div>
```



HTML 5

<nav>Menu de Navegação**</nav>**

<header>Cabeçalho**</header>**

<section>

<aside>Barra Lateral**</aside>**

<article>Conteúdo**</article>**

</section>

<footer>Rodapé**</footer>**



section

- devemos atribuir uma largura de 760px para esta tag.
- vamos centralizar esta tag. Isto é feito atribuindo suas margens para auto. Quando as margens esquerda e direita são atribuídas para auto, elas se igualam em ambos os lados e a div fica no centro do contêiner.

```
section {  
    width: 760px;  
    margin: auto;  
    background: red;  
}
```



nav

- Remova o fundo vermelho da #page-container, e adicione uma nova regra para #main- nav. Mude o fundo da #main-nav para vermelho para que possamos vê-la, e atribua sua altura para 50px:

```
article {  
  width: 760px;  
  margin: auto;  
  background: yellow;  
  float: left;  
}
```

```
nav {  
  background: red;  
  height: 50px;  
}
```



Propriedade: CLEAR

- Isso não é o que queremos. Nós queremos que o conteúdo da div seja distribuído através da div sidebar, a qual tem seu lado direito contra o lado esquerdo da barra lateral.
- Um jeito fácil de conseguir isto num layout flutuante como esse, é colocar a propriedade que limpa os floats, pois o o article e o aside estão 'flutuadas', no footer coloque:
clear: both;



CLEAR: BOTH

- Quando um elemento tem sua propriedade clearatribuída, se ele entrar em contato com um elemento flutuante, ele é colocado exatamente abaixo de onde o flutuante termina.
- Você pode especificar se ele é afetado somente por flutuadores esquerdos ou direitos, neste caso nós poderíamos utilizar tanto right ou both. Nós vamos utilizar clear: both somente por segurança.



header

- Note que nós não especificamos a largura da div. Isto por que, por padrão, a div vai se esticar até preencher seu contêiner-pai, que neste caso, é nosso body que foi configurado para 760px de largura.
- Faça a mesma coisa para a div cabeçalho (header), utilizando a altura que usamos no primeiro passo. Use a cor azul para o fundo.

```
header {  
    background: blue; height:  
    150px;  
    clear: both;  
}
```



footer

- Já que estamos aqui, vamos configurar o rodapé (footer). Use a cor laranja para o fundo.
- Lembre-se de que quando escrever sua folha-de-estilos, você pode agrupá-los. Então todos os estilos do cabeçalho (header) poderiam vir juntos. Todos os estilos do rodapé (footer) poderiam estar juntos, e os estilos de navegação poderiam estar juntos. Eu também acho que estruturar a folha-de-estilos similar à ordem do HTML ajuda bastante. Assim sendo, header fica próximo ao topo, footer ao final.

```
footer {  
    background: orange;  
    height: 66px;  
    clear: both;  
}
```



Floats

(flutuadores)

- **Floats** pode ser um conceito enganoso para fazer você perder a cabeça.
- Basicamente um float é um elemento que é alinhado contra os lados esquerdo ou direito do seu contêiner.
- No caso deste website, nós vamos fazer flutuar (float) nossa tag **aside** à direita, com a largura de 280px.

```
aside {  
  float: right;  
  width: 280px;  
  background: darkgreen;  
}
```

Adicionando conteúdo

- Somente para o propósito de testes, substitua o texto do conteúdo de article para o seguinte:

`<article>`

Texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto.

`</article>`

A interface...

[Principal](#) [Sobre](#) [Serviços](#) [Portfólio](#) [Contato](#)

Header

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto texto texto texto.

Texto texto texto texto texto texto texto texto texto texto
o texto texto texto texto texto texto texto texto texto texto
texto texto.

Footer



Menu <nav>

- Trocar o texto para:

<nav>

`Principal`

`Sobre`

`Serviços`

`Portfólio`

`Contato`

</nav>



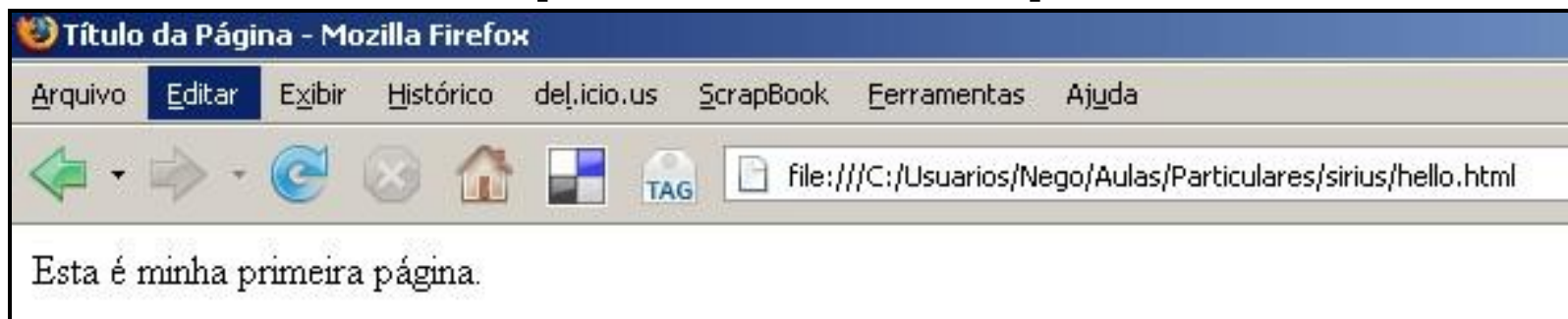
JavaScript

JavaScript

- É a linguagem de script utilizada por milhões de páginas web.
- Foi projetada para aumentar interatividade das páginas web:
 - Validação de formulários, interação com o usuário (p.ex., tratamento de cliques de botões), detecção de navegadores, etc.
- É reconhecida pela maioria dos navegadores.
- **Seu processamento é feito na máquina cliente (browser).**
- **Não há relação com Java**



JavaScript - Exemplo



```
<html>
  <head>
    <title>Título da Página</title>
  </head>
  <body>
    <script type="text/javascript">
      document.write("Esta é minha primeira página.");
    </script>
  </body>
</html>
```

JavaScript – Onde ocorrem

- Uma tag **<script/>** pode ser definida numa seção head, numa seção body e também pode ser definida externamente:
 - Na seção **head**, os scripts são **executados quando são chamados** ou quando algum evento ocorre;
 - Na seção **body**, os scripts são **executados na carga da página web**;
 - Para definição externa, um arquivo “.js” precisa ser fornecido com as funções necessárias

JavaScript – Exemplo

```
<html>
  <head>
    <title>Título da Página</title>
    <script src="hello.js"></script>
  </head>
  <body onload="message()">

    <script type="text/javascript">
      document.write("<h1>Esta é minha primeira página.</h1>");
    </script>

  </body>
</html>
```

JavaScript - Sintaxe

- Possui construções existentes na maioria das linguagens de programação (sintaxe similar a C):
 - Declaração de variáveis (*var x;*)
 - Comandos condicionais, repetições, definição de funções de usuário
 - Operadores de atribuição, comparação, ...

JavaScript

Janelas Popup

- Funções para criação de janelas popup:
 - Alerta (*alert("Texto a ser exibido");*)
 - Confirmação (*confirm("Texto a ser exibido em janela OK/Cancel");*)
 - Entrada de dado (*prompt("Label do campo de entrada", "valor padrão");*)
- Janelas que possuem o botão Cancel, quando acionado retorna valor *null*

JavaScript

Janelas Popup

```
<body>  
  <script type="text/javascript">  
    var nome = prompt("Seu nome");  
    if (nome != null && nome != "")  
      document.write("Oi " + nome);  
    else  
      document.write("Oi anônimo!");  
  </script>  
</body>
```



JavaScript – Eventos

- Eventos são ações que podem ser detectadas por um script
- Exemplos de eventos:
 - Clique do mouse, abertura de uma página web ou imagem, envio de um formulário html, uma tecla pressionada, etc
- O tratamento destes eventos pode ser a chamada de funções do script



JavaScript – Eventos

```
<html>
  <head>
    <script src="event.js"></script>
  </head>
  <body>
    
  </body>
</html>
```

- Lista de eventos:
http://www.w3schools.com/jsref/jsref_events.asp



JavaScript – Eventos

```
/* Conteúdo do arquivo event.js */  
  
function mouseOver()  
{  
    document.imagem.src = "quadrado2.gif";  
}  
function mouseOut()  
{  
    document.imagem.src = "quadrado1.gif";  
}
```

- Lista de eventos:
http://www.w3schools.com/jsref/jsref_events.asp



JavaScript – Objetos

- **JavaScript é uma linguagem OO**
- Com isto, algumas classes utilitárias padrões estão disponíveis, as quais possuem métodos e propriedades:
 - String: manipulação de strings no script

/* Exemplo de função de script que calcula o tamanho de uma string */

```
function tamanho(msg)
{
  return msg.length;
}
```

- http://www.w3schools.com/jsref/jsref_obj_string.asp



JavaScript – Objetos

- Outras classes disponíveis na linguagem JavaScript:
 - Array: armazenamento de conjuntos de valores
 - http://w3schools.sinsixx.com/js/js_obj_array.asp.htm
 - Boolean: manipulação de valores booleanos
 - https://www.w3schools.com/jsref/jsref_obj_boolean.asp
 - Math: manipulação de valores com operações matemáticas
 - http://www.w3schools.com/jsref/jsref_obj_math.asp

JavaScript – Objetos Exemplo

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>Clique no botão</p>
```

```
<button onclick="myFunction()"> Clique aqui </button>
```

```
<script>
```

```
function myFunction() {
```

```
var d=new Date();
```

```
var dias=new Array(7);
```

```
dias[0]="Domingo";
```

```
dias[1]="Segunda";
```

```
dias[2]="Terça";
```

```
dias[3]="Quarta";
```

```
dias[4]="Quinta";
```

```
dias[5]="Sexta";
```

```
dias[6]="Sábado";
```

```
document.write("Hoje é " + d.getDay());
```

```
document.write("Hoje é " + dias[d.getDay()]);
```

```
document.write("Dia qualquer: " +dias[Math.round(Math.random()*6)]);
```

```
document.write(navigator.appCodeName);
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

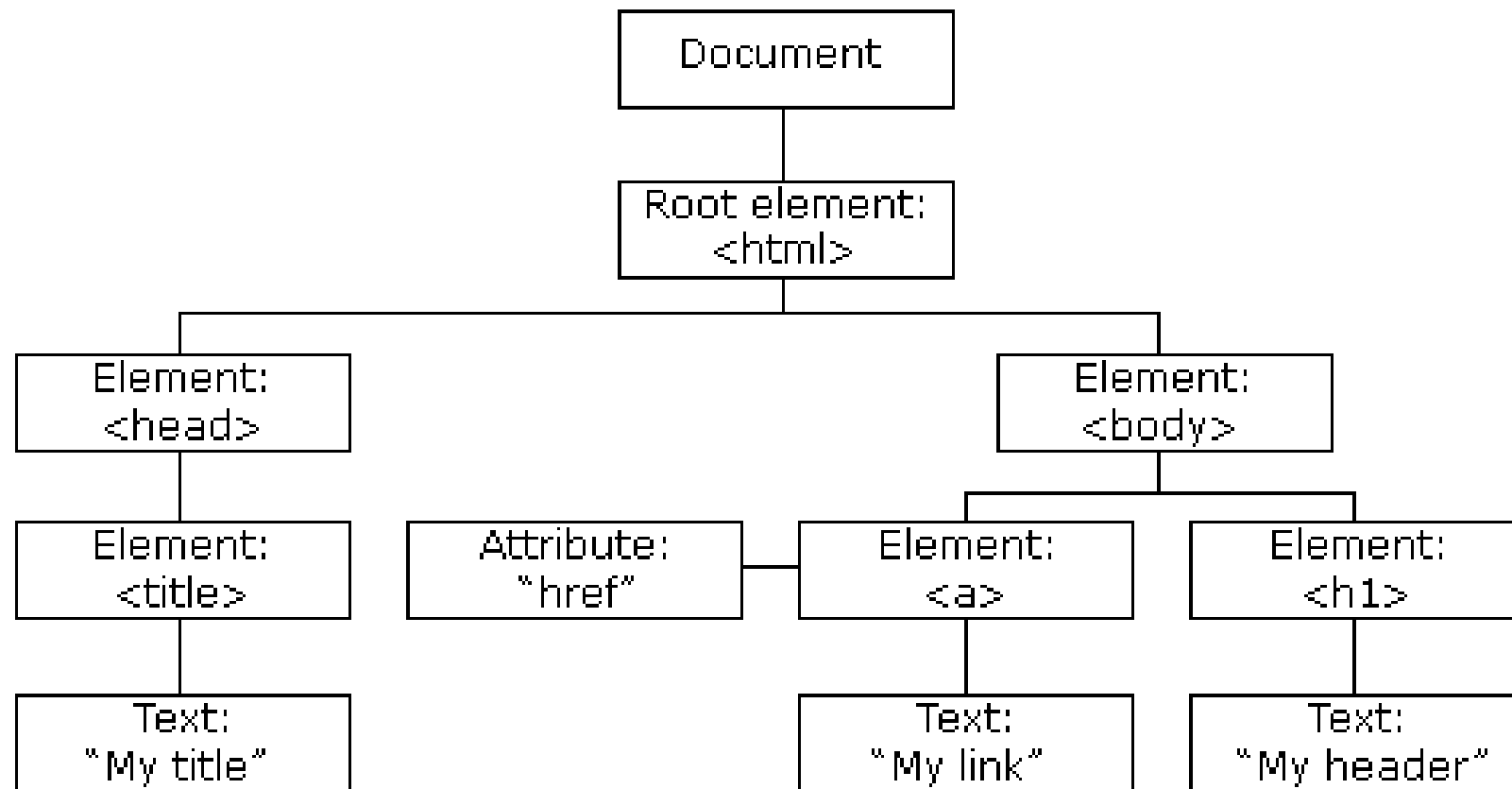


JavaScript – Objetos

- Além destas, todos os objetos HTML DOM podem ser acessados através de scripts
 - Window: objeto no topo da hierarquia do JavaScript; representa a janela do browser
 - Navigator: contém informação sobre o browser do cliente
 - Screen: contém informação sobre a tela
 - History: contém as URLs visitadas
 - Location: contém informação sobre a URL corrente
 - https://www.w3schools.com/js/js_htmlDOM.asp



JavaScript – Objetos





JavaScript – Objetos

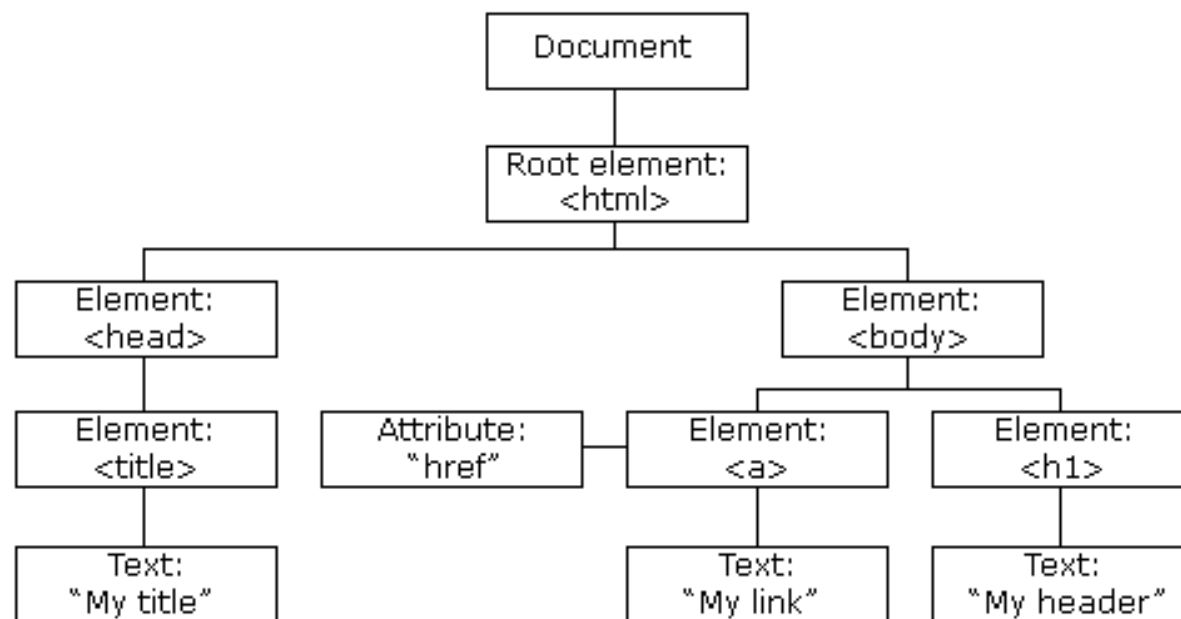
Exemplo com Window

```
<html>
<head>
  <script type="text/javascript">
    function currLocation() {
      alert(window.location);
    }
    function newLocation() {
      window.location="http://www.w3schools.com";
    }
  </script>
</head>
<body>
  <input type="button" onclick="currLocation()" value="Show current URL">
  <input type="button" onclick="newLocation()" value="Change URL">
</body>
</html>
```

HTML DOM

- Define um padrão para acesso a elementos HTML
- O DOM apresenta um documento HTML como uma estrutura em árvore

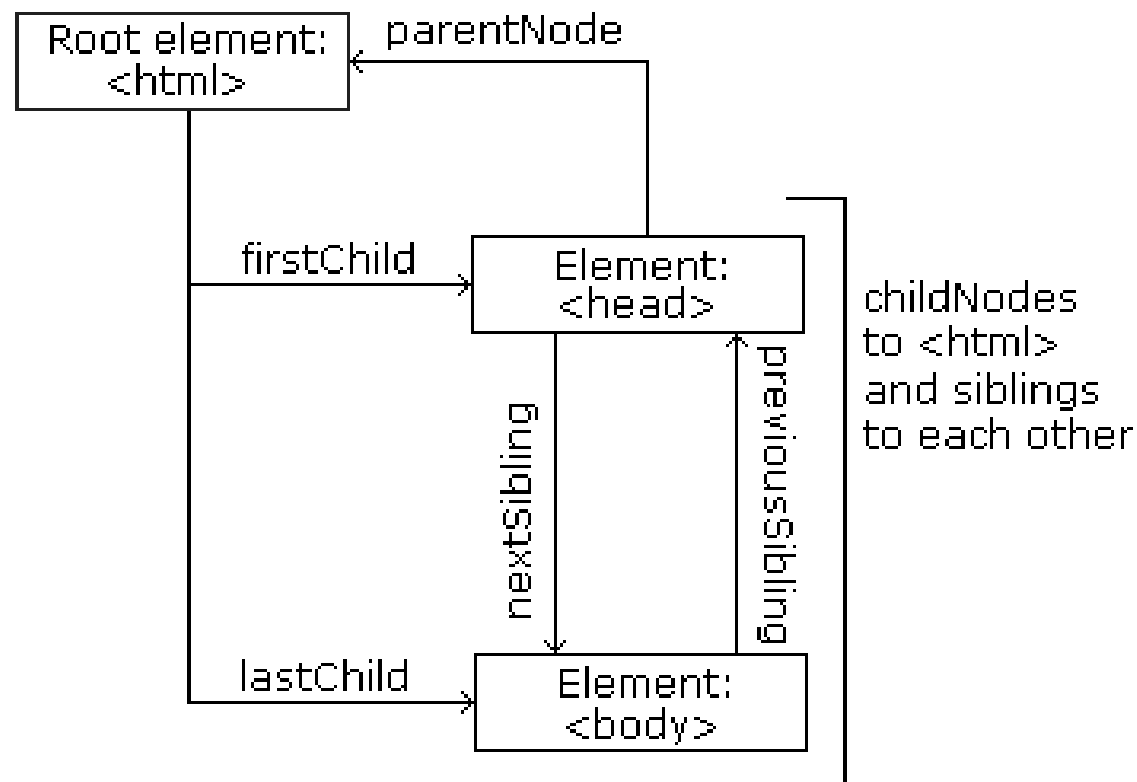
```
<html>
<head>
<title>My title</title>
</head>
<body>
  <a href="">My link</a>
  <h1>My header</h1>
</body>
</html>
```





HTML DOM

- Relacionamento entre nós numa árvore DOM





HTML DOM - API

- É definida por um conjunto de propriedades e métodos
- Algumas propriedades DOM
 - x.**innerHTML**: o valor text de x
 - x.**nodeName**: o nome do elemento x
 - x.**nodeValue**: o valor de x
 - x.**nodeType**: o tipo de x (*1 – elemento; 2 – atributo; 3 – texto; ...*)
 - x.**parentNode**: o nó pai de x
 - x.**childNodes**: os nós filhos de x
 - x.**attributes**: os nós atributos de x



HTML DOM - API

- Alguns métodos DOM
 - x.**getElementById(*id*)**: obtém o elemento com o *id* fornecido
 - x.**getElementsByTagName(*name*)**: obtém todos os elementos com a tag *name*
 - x.**appendChild(*node*)**: insere um nó filho *node* em x
 - x.**removeChild(*node*)**: remove o nó filho *node* de x

HTML DOM – API

Exemplo

```
<html>
<body>
  <p id="intro">Hello World!</p>

  <script type="text/javascript">
    txt=document.getElementById("intro").innerHTML;
    document.write("<p>Texto do parágrafo intro: " + txt + "</p>");
  </script>
</body>
</html>
```

HTML DOM – API

Exemplo

```
<html>
<body>
  <p id="intro">Hello World!</p>

  <script type="text/javascript">
    txt=document.getElementById("intro").childNodes[0].nodeValue;
    document.write("<p>Texto do parágrafo intro: " + txt + "</p>");

  </script>
</body>
</html>
```



JavaScript – Mais Usos

- Criação de cookies
- Validação de entrada
- Disparo de funções com retardo (tempo)
- Criação de objetos próprios



Referências

- <http://www.w3schools.com/>
 - Site com tutoriais on-line rápidos e com muita qualidade
- <http://www.csszengarden.com/>
 - Site que demonstra as potencialidades de CSS
- <http://del.icio.us/carlosbazilio/{css+html}>
 - Meus favoritos sobre o assunto
- <http://www.w3.org/>
 - Site do consórcio W3C