

Programação Web Front-End

HTML : Introdução, tags básicas, tags aninhadas, atributos e cores

HTML

- **HTML** (HyperText Markup Language):
linguagem de marcação de hipertexto
- **Linguagem:**
 - Bastante utilizada para exibir informações na Web.
 - Interpretada pelo navegador
 - Declarativa: não há comandos de decisão, de repetição, declaração de variáveis e de tipos...
 - A finalidade é definir a estrutura e conteúdo de um arquivo (texto)

HTML

- Marcação:

Imagens



By N2H

Google

Error

Cores de fonte

We're sorry...

Título destacado

... but your query looks similar to automated requests from a computer virus or spyware application. To protect our users, we can't process your request right now.

We'll restore your access as quickly as possible, so try again soon. In the meantime, if you suspect that your computer or network has been infected, you might want to run a [virus checker](#) or [spyware remover](#) to make sure that your systems are free of viruses and other spurious software.

If you're continually receiving this error, you may be able to resolve the problem by deleting your Google cookie and revisiting Google. For browser-specific instructions, please consult your browser's online support center.

If your entire network is affected, more information is available in the [Google Web Search Help Center](#).

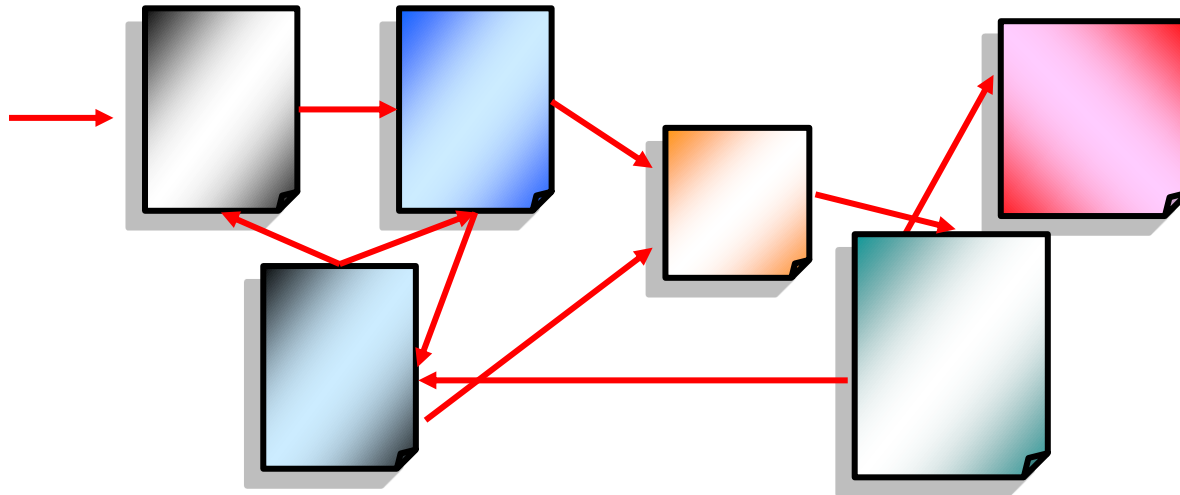
We apologize for the inconvenience, and hope we'll see you again on Google.

Régua horizontal

Link

HTML

- **Hipertexto:**
 - é um tipo de texto que possui ligações/referências entre si



- Documentos/arquivos criados em HTML:
 - São chamados de páginas
 - Devem ter extensão **.html**
 - Pode apresentar apenas texto ou ainda uma combinação de texto, imagens e links (dentre outros).

HTML

- **Marcação** (cont.):

- Indica partes especiais em uma página.
Ex: partes em **negrito**, partes em *itálico*, inserção de imagens e etc.
- É feita através de comandos denominados tags ou descritores
- Ao ler um documento HTML, o navegador tenta interpretar todas as seqüências de caracteres que ficam entre os símbolos "<" e ">"
- Qualquer coisa que estiver entre esses caracteres faz parte da tag e não deve ser mostrado na tela.



Ex: Um texto entre as tags **** **** é exibido no browser em negrito

**** Teste de tag **** → **Teste de tag**

HTML

- **Marcação** (cont.):
 - Tags duplas apresentam abertura e fechamento:
`<tag>Texto a ser formatado</tag>`
 - Tags simples apresentam apenas uma tag:
`<tag/>`
 - Tags podem ter ainda outros elementos chamados atributos:
`<tag atributo1="valor1" atributo2="valor2" ... >Texto a ser formatado</tag>`
`<tag atributo1="valor1" atributo2="valor2" ... />`

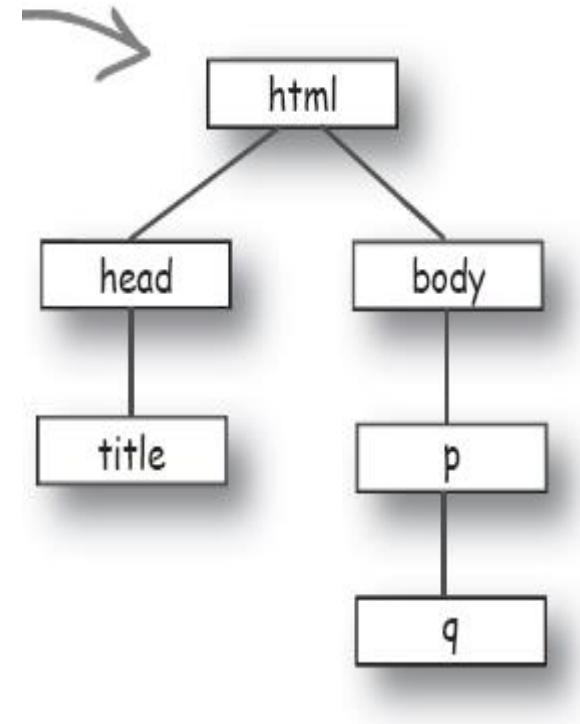
Algumas tags

- `<!-- comentário -->`
- ``: **negrito**
- ``: *itálico*
- `<u></u>`: sublinhado
- `
`: quebra de linha (break row)
- `<hr/>`: linha horizontal (horizontal rule)
- `<h1></h1>` , `<h2></h2>` , ... , `<h6></h6>`: cabeçalho (heading).
 - Dão destaque aos textos com fontes maiores.
 - Quanto maior o número, menor o tamanho do texto
- `<p></p>`: parágrafo (paragraph).
 - Usado para demarcar parágrafos em um texto.

Obs.: A quebra de linha nas tags `<h1>` e `<p>` são maiores que na tag `
`

Estrutura básica HTML

- A página HTML anterior está estruturalmente errada, pois não obedece à estrutura básica definida no HTML
- Estrutura básica de uma página:
 - **html**: informa que a página contém código da linguagem html
 - **head**: cabeçalho que contém identificações de uma página
 - **title**: título da página, aparece na barra de títulos e nas abas do navegador
 - **body**: contém corpo da página. Nele aparece todo o conteúdo como texto, links, imagens, etc



Estrutura básica HTML

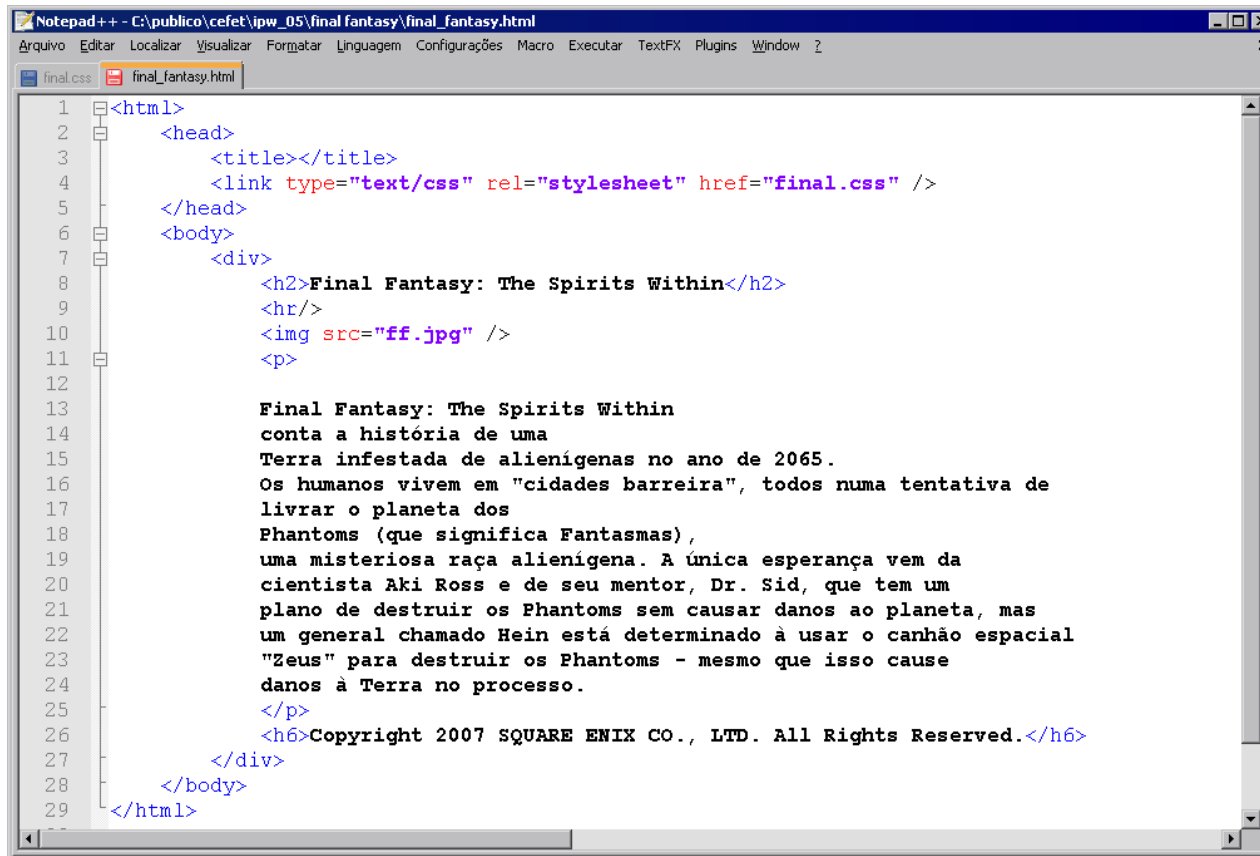
- Toda página deve possuir a estrutura básica
- A ordem e aninhamento das tags devem ser obedecidas
 - Todas as tags da estrutura básica **são duplas**
 - **html** é a tag raíz e dentro dela estão todas as outras
 - Dentro da tag **head** temos a tag **title**.
 - Na tag **title** não podemos colocar mais nada a não ser texto (sem outras tags)
 - O conteúdo visível da página deve ser colocado dentro da tag **body**

Dicas

- Ao abrir uma tag dupla, feche-a, assim haverá menos risco de esquecê-la aberta
- Salve sempre seu trabalho
- Antes de usar a complementação do editor ou recursos avançados de um editor, procure aprender e memorizar as principais tags
- SEMPRE faça backup de seu trabalho, pois o mesmo pode ser apagado
- Tenha, sempre que possível, modelos para não começar o trabalho do zero
- Sempre repita os exercícios que não ficaram bem fixados

HTML

- Para criarmos documentos HTML, podemos usar qualquer editor de texto (inclusive o bloco de notas)
- Usaremos o **Notepad++** ou **Sublime**:



```
1 <html>
2   <head>
3     <title></title>
4     <link type="text/css" rel="stylesheet" href="final.css" />
5   </head>
6   <body>
7     <div>
8       <h2>Final Fantasy: The Spirits Within</h2>
9       <hr/>
10      
11      <p>
12
13        Final Fantasy: The Spirits Within
14        conta a história de uma
15        Terra infestada de alienígenas no ano de 2065.
16        Os humanos vivem em "cidades barreira", todos numa tentativa de
17        livrar o planeta dos
18        Phantoms (que significa Fantasmas),
19        uma misteriosa raça alienígena. A única esperança vem da
20        cientista Aki Ross e de seu mentor, Dr. Sid, que tem um
21        plano de destruir os Phantoms sem causar danos ao planeta, mas
22        um general chamado Hein está determinado a usar o canhão espacial
23        "Zeus" para destruir os Phantoms - mesmo que isso cause
24        danos à Terra no processo.
25      </p>
26      <h6>Copyright 2007 SQUARE ENIX CO., LTD. All Rights Reserved.</h6>
27    </div>
28  </body>
29</html>
```

Prática 00



```
<!DOCTYPE html>
```

```
<html lang="pt-br">
```

```
<head>
```

```
<meta charset="utf-8"/>
```

```
<title>Alo Mundo !!!</title>
```

```
</head>
```

```
<body>
```

```
Alo Mundo !!! <br/>Regua horizontal
```

```
<hr/>
```

```
<strong>Seu nome</strong><br/>
```

```
</body>
```

```
</html>
```

Arquivo:
alomundo.html

Título:
Alô Mundo !!!

Prática 01

- Construa uma página HTML com as instruções abaixo:
 - Nome do arquivo: dados.html
 - Título: Dados pessoais
 - No corpo, deve haver as seguintes informações:

Dados Pessoais

Nome: _____

Data de nascimento: _____

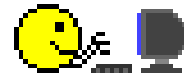
Cidade natal: _____

Estado: _____

E-mail: _____

Outras informações: _____

- Realce com negrito os trechos “Nome:”, “Data de Nascimento”, etc...
- Quebre a linha a cada nova informação



Exame de vista ← h1

Leia apenas o que se pede: ← h2

hr

números : a 2 3 u 5 j 8 ← h3

letras : y a l 2 / 3 q ← h4

vogais : l a + o p 3 i ← h5

consoantes : c a 8 o # d i ← h6

Nome: exame.html

Título: Exame de vista

Tags aninhadas

- Podemos ter mais de uma tag formatando um mesmo objeto:

`<p><u>U2</u> - Famosa banda irlandesa</p>`

- Os blocos de texto em HTML podem conter outros blocos. O bloco mais interno deverá ser fechado antes do bloco externo:

`<h1>Texto<u>muitoimportante</u>para todos</h1>`

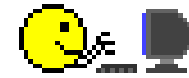
- De forma bem popular, dizemos que fechamos as tags “de dentro pra fora”

- Nunca use construções como esta:

`<h1>Texto<u>muito</h1>importante</u>para todos`



Prática 03



Algumas traduções e significados

Access: software que permite manipular dados. Não é um banco de dados

HTML: linguagem de criação de páginas.

Procedure: trecho de código em Pascal.

Ruby: linguagem de programação com tipagem dinâmica.

Tags: elementos da linguagem HTML.

Nome: dicionario.html

Título: Dicionário de informática

Atributos

- Atributos fornecem mais opções de formatação a algumas tags
- As tags podem ter um ou mais atributos que modificam alguma propriedade do texto marcado:

`<h1 align="center">Texto Centralizado</h1>`

- Os atributos são pares do tipo:

`propriedade="valor"`

- A ordem deles não faz diferença, mas sempre estão na tag de inicial ou de abertura
- Se um atributo não for reconhecido pelo browser, será ignorado

Alguns atributos

- **<hr/>**

- width: comprimento. Pode ser um valor numérico ou %
- size: espessura. Pode ser um valor numérico ou %
- color: cor. Valores RGB (**Red Green Blue**)
- align: alinhamento. Valores center, left, e right

Ex: `<hr width="50%" color="#0000ff" align="right"/>`
`<hr width="250" color="red" align="center"/>`

- **<body></body>**

- bgcolor: cor do fundo da página. Valores RGB (acima)
- text: cor do texto da página. Valores RGB (acima)

Ex: `<body bgcolor="darkgray" text="white">...</body>`

- **<p></p>, <hn></hn>**

- align: alinhamento. Valores: center, left, right e justify

Cores

- Em HTML usamos uma combinação numérica hexadecimal para expressar as cores: “#rrggbb”
 - Usa-se um sustenido (sharp #) antes dos 3 pares de números
 - O primeiro par é pra cor **vermelha**, em seguida outro para a cor **verde** e um último para a cor **azul**
 - Para exibir uma cor, deve-se atribuir um valor diferente de **00**
 - O máximo de uma cor é definido atribuindo-se o valor **ff** (15)
 - Tons intermediários são obtidos combinando-se os valores dos 3 pares de cores

Ex:

- cor azul - cores vermelha e verde “zeradas” e azul no máximo (ff):
`<hr color="#0000ff"/>`
- Cor acinzentada:
`<hr color="#808080"/>`

Cores

- Algumas cores possuem constantes para facilitar o uso:
aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow ...
- Link com a listagem de cores no site do W3C:
http://www.w3schools.com/colors/colors_names.asp
- Cores por nome: há uma limitação na quantidade de nomes
- Cores por número: mais difícil de lembrar, porém com mais opções

aqua (#00ffff)	black (#000000)	blue (#0000ff)	fuchsia (#ff00ff)
green (#008000)	gray (#808080)	lime (#00ff00)	maroon (#800000)
navy (#000080)	olive (#808000)	purple (#800080)	red (#ff0000)
silver (#c0c0c0)	teal (#008080)	white (#ffffff)	yellow (#ffff00)

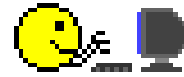
- Dica: não abuse nas cores

Prática 04 e Prática 05



- Altere as práticas 02 (exame.html) e 03 (dicionario.html) da seguinte forma:
 - Cor do texto: darkblue
 - Cor do fundo: #dddddd
 - Cor das réguas horizontais: darkblue
- **Obs:**
 - Salve os arquivos alterados com outro nome
 - Após completar atividade, verifique outras opções de valores dos atributos

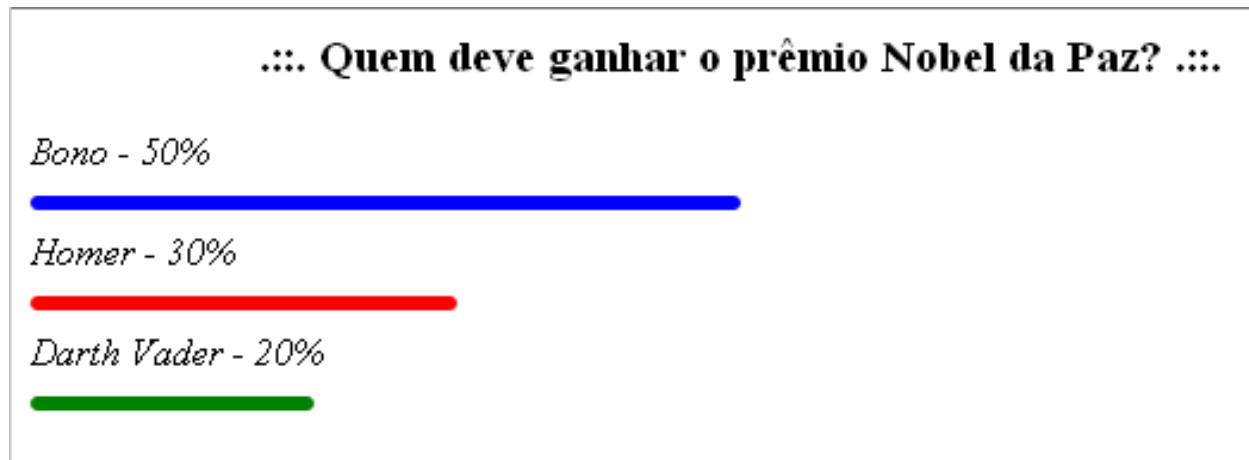
Prática 06

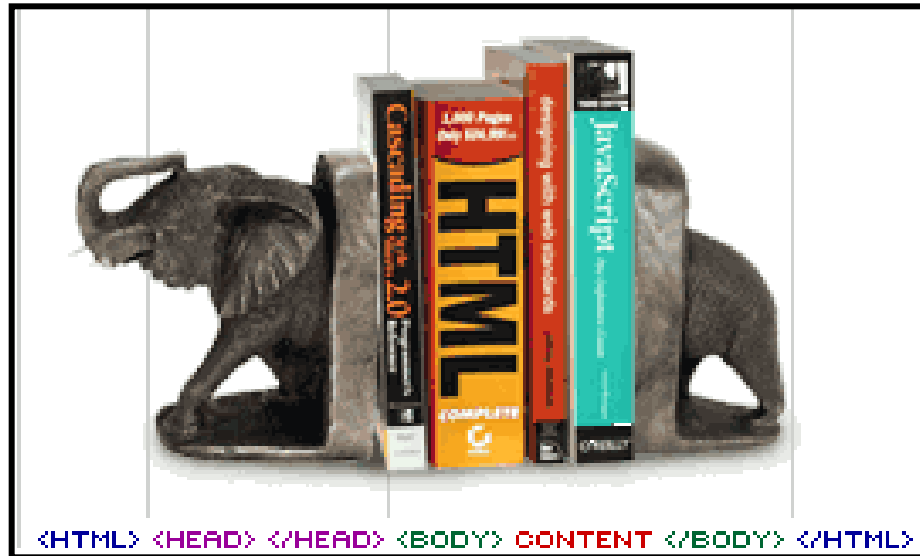


- Crie uma página que simule uma enquete semelhante à ilustração abaixo:

– Notas:

- O título é uma tag `<h3></h3>` centralizada
- a barra das opções são tags `<hr/>` com os atributos de cor, comprimento e espessura alterados.
- Nome do arquivo: enquete.html.
- Título : Enquete do Premio Nobel da Paz





Programação Web Front-End

HTML: fontes, listas, imagem, audio e video

Fontes

- ****: permite formatar o tipo, tamanho e a cor da fonte
- Atributos:
 - **face**: tipo da fonte. Mais comuns: arial, helvetica, verdana, times, serif, sans-serif, futura, cursive, fantasy, monospace, georgia.
 - **size**: tamanho. Varia de 1 a 7 ou de -3 a 3
 - **color**: cor definida em hexadecimal no padrão rgb (vermelho, verde e azul: **#2233ff**) ou definida pelo nome (red, blue, black)

Nota: formatar texto bem como outras formatações via atributos está em desuso com o uso de folhas de estilo (CSS)

Fontes

- Ex:

` Texto a ser formatado `

` Texto a ser formatado`

- Dicas:

- use fontes conhecidas, pois nem todo “cliente” pode ter as mesmas fontes que você
- Não abuse do uso de fontes diversas assim como o uso de cores
- Caso seja indispensável o uso de uma fonte menos conhecida, gere imagens com o texto desejado

Prática 07

- Crie o logotipo do Google usando formatação de fontes:
 - Tipo: Arial
 - Tamanho: 7
 - Cores: variadas

The image shows the Google logo, which consists of the word "Google" in a sans-serif font. The letters are colored: 'G' is blue, the first 'o' is red, the second 'o' is yellow, 'g' is blue, 'l' is green, and 'e' is red.

Prática 08



Pizza

← h2 e fonte arial

Todas as fontes e <hr/> da página com cor vermelha

Pizza é uma preparação culinária que consiste em um disco de massa fermentada de farinha de trigo, regado com molho de tomates e coberto com ingredientes variados que normalmente incluem algum tipo de queijo, carnes preparadas ou defumadas e ervas, normalmente orégano ou manjericão, tudo assado em forno.

Alguns tipos e sabores

← h3 e fonte arial

A variedade de coberturas que se pode colocar sobre uma pizza é quase infinita, entretanto, algumas preparações são tradicionais e têm fiéis seguidores:

Margherita
Mussarela
Portuguesa
Calabresa
Toscana
Pepperoni
Quatro queijos
Pomodoro

← fonte verdana e tamanho 2

Fontes

<http://pt.wikipedia.org/wiki/Pizza>
<http://www.pizza.it>
http://en.wikipedia.org/wiki/History_of_pizza

← fonte verdana e tamanho 1

www.pizzaonline.com.br ←

Listas

- Há 3 tipos de listas em HTML:

Listas não ordenadas

- ◆ Maçã
- ◆ Banana
- ◆ Uva
- ◆ Abacate

Listas ordenadas

1. Maçã
2. Banana
3. Uva
4. Abacate

Listas de definições:

Maçã

A maçã é excelente para prevenir e manter a taxa de colesterol

Banana

É uma fonte de vitamina A,C, fibras e potássio.

Uva

A uva é o fruto da videira, da família das Vitaceae

Abacate

O abacate é o fruto comestível do abacateiro

Listas

- Listas são blocos em que podemos exibir itens agrupados
- Os itens podem ser textos, parágrafos, imagens, outras listas...
- `...`: listas não ordenadas (**u**nordered **l**ist):
 ``
 `Item 1`
 `Item 2`
 `Item 3`
 `...`
 ``
- Atributo `type` especifica o formato do marcador.
Valores: “circle”, “square”, “disc”

Listas

- `...`: listas ordenadas (ordered list):
 ``
 `Item 1`
 `Item 2`
 `Item 3`
 `...`
 ``
- Atributo type: indica o tipo de ordenação. Valores:
 - 1: número arábico
 - a: letras minúsculas,
 A: letras maiúsculas,
 - i: números romanos minúsculos
 I: números romanos maiúsculos
- Atributo start: indica o “número inicial”
- Atributo reversed: em ordem decrescente

Listas

- Exemplos:

```
<ul type = "circle">  
  <li>Maçã</li>  
  <li>Banana</li>  
  <li>Uva</li>  
  <li>Abacate</li>  
</ul>
```



- Maçã
- Banana
- Uva
- Abacate

```
<ol type = "A">  
  <li>Maçã</li>  
  <li>Banana</li>  
  <li>Uva</li>  
  <li>Abacate</li>  
</ol>
```



- A. Maçã
- B. Banana
- C. Uva
- D. Abacate

Listas

- `<dl>...</dl>`: lista de definição (definition list)

`<dl>`

`<dt>Termo 1</dt>`

definition term

`<dd>Definição do termo 1</dd>`

definition
“description”

`<dt>Termo 2</dt>`

`<dd>Definição do termo 2</dd>`

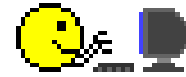
`<dt>Termo 3</dt>`

`<dd>Definição do termo 3</dd>`

...

`</dl>`

Prática 09



1. Modifique a página da Prática (pizza.html) para utilizar listas não ordenadas nos nomes das pizzas e listas ordenadas nos 3 links finais.

- ♦ Margherita
- ♦ Mussarela
- ♦ Portuguesa
- ♦ Calabresa
- ♦ Toscana
- ♦ Pepperoni
- ♦ Quatro queijos
- ♦ Pomodoro

1. <http://pt.wikipedia.org/wiki/Pizza>
2. <http://www.pizza.it>
3. http://en.wikipedia.org/wiki/History_of_pizza

Nota: *quando se usa listas, os itens já ficam com quebra de página automática. Portanto, retire quebras de linha desnecessárias.*

Prática 10

- Refaça a atividade do dicionário usando listas de definição

Algunas traduções e significados

Access:

software que permite manipular dados. Não é um banco de dados

HTML:

linguagem de criação de páginas.

Procedure:

trecho de código em Pascal.

Ruby:

linguagem de programação com tipagem dinâmica.

Tags:

elementos da linguagem HTML.



Imagens



- Adicionar uma imagem a uma página HTML é simplesmente **referenciar** uma imagem em arquivo
- A imagem não fica dentro do arquivo HTML e sim o endereço/caminho
- Formatos de imagem mais comuns na WEB:
 - **GIF** (Graphics Interchange Format): melhores para gráficos sem muitas cores (até 256) ou figuras animadas
 - **JPEG** (Joint Photographic Experts Group): melhores para fotografia
 - **PNG** (Portable Network Graphics): melhor que os formatos JPEG e GIF: milhões de cores e efetiva compressão, mas ainda não tão popular
- Leitura adicional:
<http://www.infowester.com/imagensnet.php>

Imagens

- ``: O atributo **src** é obrigatório e representa o local de onde “carregar” a imagem
- Uma imagem também pode ser carregada no fundo de uma página:

...`<body background="caminho_imagem">...</body>`

- Caminho absoluto: é o caminho/endereço completo para se chegar à imagem

``

``

``



Imagens

- Caminho relativo: é o caminho/endereço parcial em relação à página para se chegar à imagem.
- Há 3 situações para caminhos relativos
 - Imagem no mesmo diretório da página: o caminho será apenas o nome da imagem:
``
 - Imagem em um diretório a partir do diretório da página: o caminho será o nome do diretório + "/" + o nome da imagem:
``
 - Imagem em um diretório anterior/"acima" do diretório da página: utiliza-se ".." para voltar um diretório:
``

Imagens

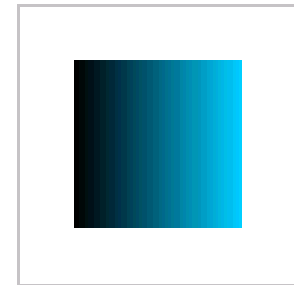
O arquivo “index.html” está na pasta “pagina”:

```
...  
<img src=“quadrado_verde.jpg”/>
```

```
...  
<img src=“imagens/quadrado_vermelho.jpg”/>
```

```
...  
<img src=“../quadrado_azul.jpg”/>
```

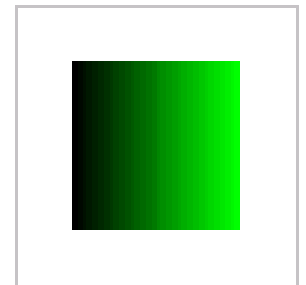
```
...
```



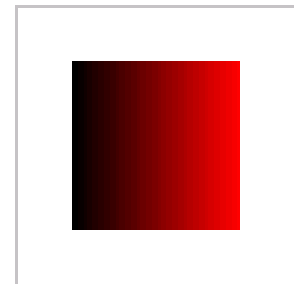
quadrado_azul.jpg



index.html



quadrado_verde.jpg



quadrado_vermelho.jpg

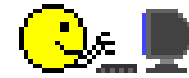
Imagens

- Atributos:
 - **align**: alinhamento
 - top, middle e bottom: alinha o texto em relação à imagem
 - left, right: alinha a imagem em relação ao que estiver em volta
 - **alt**: exibe um texto de descrição ou alternativo para a imagem.
 - **border**: especifica a largura da borda da imagem em pixels
 - **height** e **width** : altura e largura. Podem ser valores numéricos ou %
- Ex:
``

Imagens

- Dicas de uso:
 - Crie sempre um diretório para suas imagens
 - Não perca tempo à toa, observe o nome das imagens:
dog.jpg <> dog.JPG <> dog.png <> dogs.jpg <> dogs.gif
 - Não use imagens muito grandes/pesadas
 - Caso seja necessário mostrar várias imagens em uma página, mostre miniaturas das originais como links para as imagens maiores
 - Verifique se a qualidade (quantidade de cores) da imagem é a ideal. Se for o caso, dê preferência a imagens com 256 cores
 - Gifs animados são legais, mas muitos tornam o visual da página *trash*
 - Quando optar em substituir um texto por uma imagem (ex: botão de voltar) use uma imagem intuitiva e leve
 - Se sua página tem um visual simples com poucas cores prefira as imagens que tem poucas cores também.

Prática 11



Pizza

icone_pizza.gif

topo.png, largura 100%, altura 5%

Pizza é uma preparação culinária que consiste em um disco de massa fermentada de farinha de trigo, regado com molho de tomates e coberto com ingredientes variados que normalmente incluem algum tipo de queijo, carnes preparadas ou defumadas e ervas, normalmente orégano ou manjeriço, tudo assado em forno.



Alguns tipos e sabores

A variedade de coberturas que se pode colocar sobre uma pizza é quase infinita, entretanto, algumas preparações são tradicionais e têm fiéis seguidores:

- ♦  Margherita
- ♦  Mussarela
- ♦  Portuguesa
- ♦  Calabresa
- ♦  Califórnia
- ♦  Pepperoni
- ♦  Quatro queijos
- ♦  Bacon

Miniaturas (sufixo **_peq**) com:
alinhamento ao meio,
borda de 2 pixels



Fontes

1. <http://pt.wikipedia.org/wiki/Pizza>
2. <http://www.pizza.it>
3. http://en.wikipedia.org/wiki/History_of_pizza

Prática 12

- Refaça a atividade da enquete usando imagens alterando a altura e largura das figuras de 1 píxel passadas (*azul.png*, *vermelho.png* e *verde.png*).

... Quem deve ganhar o prêmio Nobel da Paz? ...

Bono - 50%



Homer - 30%



Darth Vader - 20%



Enquete encerrada em julho/2008

Audio

- Uso da tag AUDIO

a)

```
<audio controls autoplay>
```

```
  <source src="audios\horse.wav" type="audio/wav">
```

```
  <source src="audios\horse.ogg" type="audio/ogg">
```

```
  <source src="audios\horse.mp3" type="audio/mpeg">
```

Seu navegador não suporta a tag AUDIO.

```
</audio>
```

b)

```
<audio controls autoplay
```

```
  src="audios\horse.mp3" type="audio/mpeg">
```

```
</audio>
```

Video

- Uso da tag VIDEO

a)

```
<video width="320" height="240" controls>  
  <source src="videos\movie.webm" type="video/webm">  
  <source src="videos\movie.mp4" type="video/mp4">  
  <source src="videos\movie.ogv" type="video/ogg">  
  Seu navegador não suporta a tag VIDEO.  
</video>
```

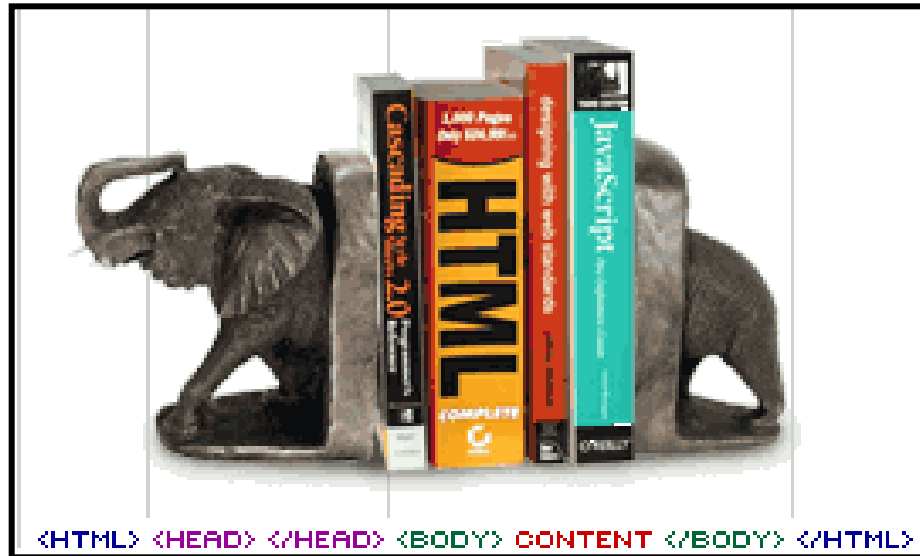
b)

```
<video width="320" height="240" controls  
  src="videos\movie.mp4" type="video/mp4">  
</video>
```

Video do Youtube

- Uso da tag OBJECT

```
<object data="http://www.youtube.com/v/xjfVDxttN7Y"  
  width="320" height="240" />  
</object>
```

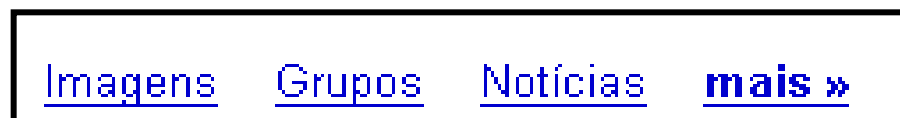


Programação Web Front-End

HTML : links, tabelas, formulários, caracteres de escape e elementos do head

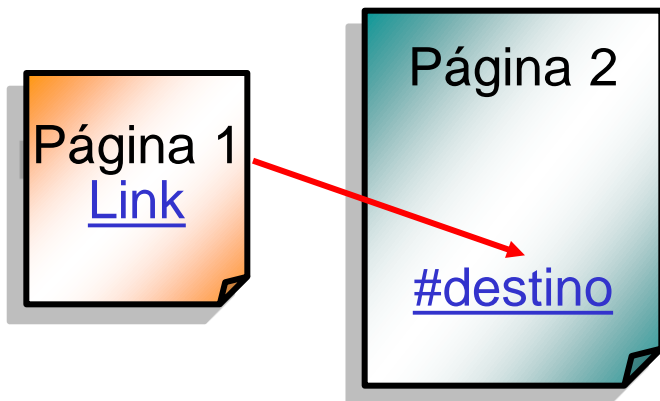
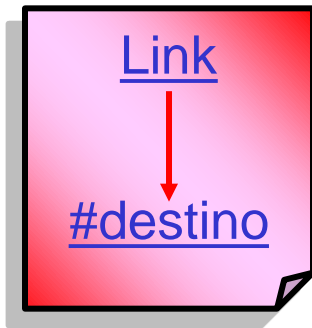
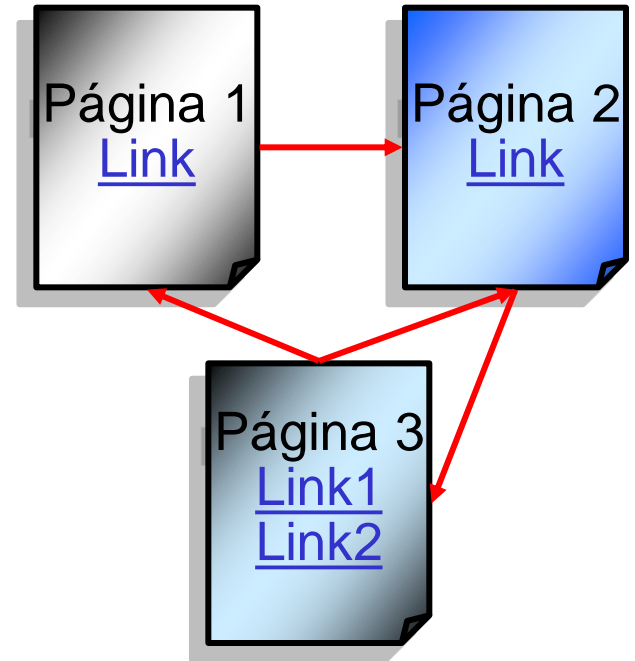
Links

- Hyperlinks ou simplesmente links conectam / vinculam / ligam / apontam páginas e outros recursos na WEB
- São a razão de ser do hipertexto, pois nos permitem “navegar” pela WEB
- Para criar links em HTML são necessárias 3 coisas:
 - O endereço ou referência para onde o link deve “apontar”
 - Algum texto ou imagem que descreva o link
 - O uso da tag `<a>...` (anchor) que cria uma ligação e pode ser “clorada”




Links

Navegação horizontal: entre páginas e/ou entre outros objetos (arquivos texto, imagens, etc).



Navegação vertical: vínculo dentro de uma página (não necessariamente a mesma).

Links

- ` Descrição `
 - `<a>..` cria uma ligação e pode ser “clorada”
 - `href` é o endereço, caminho ou referência para onde o link deve “apontar”
 - `href` semelhante ao atributo `src` da tag ``. Caso o endereço seja absoluto, lembre-se de usar sempre o `http://`
 - `target` indica onde o link será aberto
 - Entre as duas tags temos um texto de descrição. Pode ser outra construção HTML
- Ex. de links usando endereços absolutos:
 - `Green Peace`
Endereço absoluto para uma página na WEB
 - `???` ← 



Links

- Ex. de links com endereços relativos:

- `Alô mundo`

Considerando a página **alomundo.html** dentro do mesmo diretório

- `Alô mundo`

Considerando a página **alomundo.html** dentro de um diretório práticas

- `Alô mundo`

Considerando a página **alomundo.html** dentro de um diretório anterior ao diretório atual

Links

- Cores nos links: para mudar as cores padrões podemos declarar 3 atributos na tag body:
 - **link** = “cor” : cor do link normal
 - **vlink** = “cor”: cor do link visitado
 - **alink** = “cor”: cor do link ativado (clicado com o mouse em cima)
- Ex:
<body link=“blue” vlink=“red” alink= “green”>

Prática 13

- Crie no seu diretório uma página chamada **praticas.html** com links para as práticas anteriormente feitas
- Ao lado de cada link, coloque uma descrição do que foi usado para a prática em questão:

Minhas atividades práticas

[Alo mundo](#): uso das tags básicas do HTML

[Dados pessoais](#): uso de quebra de linha e negrito

...

[Enquete](#): uso fontes, réguas horizontais e atributos

...

[Pizza](#): listas, fontes e imagens

- Use sua imaginação: fontes, cores e outros são livres!

Links – Navegação Vertical

- Usando o atributo **name**, pode-se navegar verticalmente dentro de uma página
- name define um “destino” dentro de uma página
- Utilizando outro link com o href apontando para o destino definido, o browser “desloca” a página

`Endereço`

`Endereço:`


- [Dados pessoais](#)
- [Endereço](#)

Dados pessoais:

xxx

yyy

....

Endereço: 
Rua A, no. 22

Prática 14

- Crie uma página chamada **cursos.html** semelhante à página abaixo usando links com navegação vertical para navegar pelos 4 cursos e subir a página para o topo

Cursos de webdesign

Conheça nossos cursos:

- [HTML](#)
- [DreamWeaver](#)
- [Flash](#)
- [Fireworks](#)

HTML

O que é?

HTML (acrônimo para a expressão inglesa HyperText Markup Language, que significa Linguagem de Marcação de Hipertexto) é uma linguagem de marcação utilizada para produzir páginas na Web. Documentos HTML podem ser interpretados por navegadores.

Sobre o curso:

Este curso ensina desde os conceitos básicos da linguagem HTML, até a criação de formulários para a Web, ideal para iniciantes, o par quem quer um guia de consulta, também possui um Glossário com todas as palavras mais utilizadas na Internet

[sobe](#)

Dreamweaver

O que é?

O Macromedia Dreamweaver é um software de criação para a web criada pela Macromedia (agora Adobe Systems), que está atualmente na versão CS3. Versões iniciais da aplicação serviam como um simples editor HTML WYSIWYG ("What You See Is

Prática 15

- Modifique a página pizza.html para utilizar links:
 - Na seção: fontes, transforme os links enumerados em links normais
 - Na seção de tipos, faça com que cada um dos tipos seja um link para a imagem ampliada da pizza (presentes no diretório indicado).
 - Transforme em link também o endereço na parte inferior da página (<http://www.pizzaonline.com.br>)
 - Modifique as cores dos links (normal, visitado e ativado) para as cores rosa, azul e verde respectivamente

Lembre-se: é sempre uma boa prática utilizar uma pasta/diretório para as imagens

Tabelas

- Construções que permitem agrupar texto, imagem, links e até mesmo outras tabelas em suas células
- Elementos presentes em uma tabela:
 - bordas
 - linhas e colunas formando as células
 - eventualmente células expandidas
- Servem também como elemento de controle e layout em páginas web

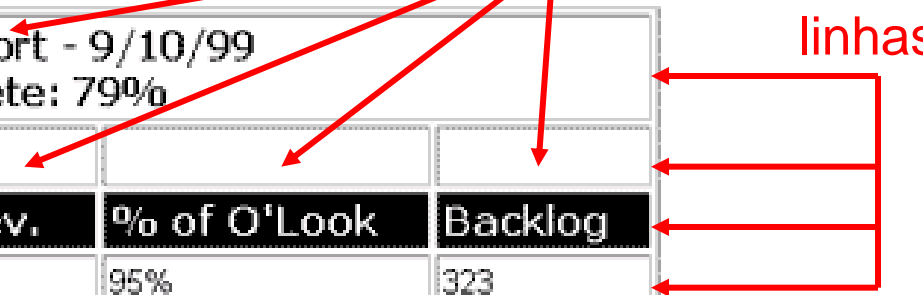


Diagram illustrating table structure with red arrows pointing to columns and rows.

SCAAL Earning Report - 9/10/99 Quarter Complete: 79%				
Summer Blend	Outlook	Net. Rev.	% of O'Look	Backlog
North America	7,317	6,932	95%	323
Commercial Distribution	3,900	4,505	116%	(12)
On-line Store/ESD	962	833	87%	7

Tabelas

- Basicamente usam-se 4 tags duplas: **table** (tabela), **th** (table header) – opcional, **tr** (table row) e **td** (table data)
- Definição hierárquica:

<table>

<th> ... </th>

<tr>

<td>

....

</td>

<td>

....

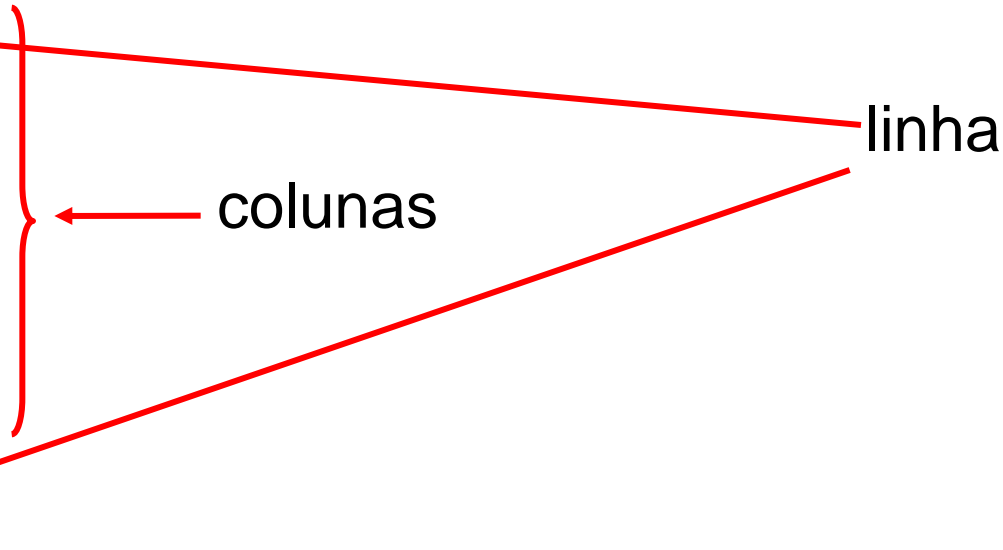
</td>

</tr>

<tr>

...
</tr>

</table>



Tabelas

- Exemplo:

```
<html>
```

```
....
```

```
<table border = "1">
```

```
  <tr>
```

```
    <th>Coluna 1</th>
```

```
    <th>Coluna 2</th>
```

```
  </tr>
```

```
  <tr>
```

```
    <td>linha 1, coluna 1</td>
```

```
    <td>linha 1, coluna 2</td>
```

```
  </tr>
```

```
  <tr>
```

```
    <td>linha 2, coluna 1</td>
```

```
    <td>linha 2, coluna 2</td>
```

```
  </tr>
```

```
</table>
```

```
....
```

```
</html>
```

Tabelas

- **<table>**: tag dupla que define a tabela. Principais atributos:
 - **bgcolor**: cor de fundo da tabela
 - **border**: largura da borda em pixels.
 - **width** e **height**: largura e altura da tabela em pixels ou %.
 - **align**: alinha a tabela em relação ao navegador. Valores: “center”, “left”, “right”
 - **background**: imagem como “preenchimento” da tabela.
 - **cellspacing**: distância em pixels entre as bordas de células adjacentes
 - **cellpadding**: distância em pixels entre a borda e o conteúdo

Tabelas


- **<tr>**: tag dupla que define uma linha dentro de **<table>**.
- Principais atributos:
 - **valign**: alinhamento vertical. Valores: “top”, “middle”, “bottom”
 - **align**, **bgcolor**, **width**, **height**, **background**: mesmos valores de **<table>**
- **<td>**: tag dupla que define uma coluna dentro de **<tr>**. Podem-se ter várias colunas dentro de uma linha.
- Principais atributos:
 - **align**, **valign**, **bgcolor**, **width**, **height**, **background**: os mesmos valores da tag **<table>**

Tabelas

- Exemplo:

...

```
<table align = "center">
  <tr>
    <td bgcolor = "#006600">
      <font color = "white"><b>Menu</b></font>
    </td>
  </tr>
  <tr>
    <td bgcolor = "#d0cc9c">Principal</td>
  </tr>
  <tr>
    <td bgcolor = "#d0cc9c">Fotos</td>
  </tr>
  <tr>
    <td bgcolor = "#d0cc9c">Links</td>
  </tr>
  <tr>
    <td bgcolor = "#d0cc9c">Fale conosco</td>
  </tr>
</table>
```



Menu
Principal
Fotos
Links
Fale conosco

...

Prática 16



- Altere a página pizza.html trocando as listas dos sabores por uma tabela com as seguintes formatações:
 - Duas linhas e quatro colunas
 - Cellspacing e cellpadding: 10 pixels
 - Alinhamento centralizado



Alguns tipos e sabores

A variedade de coberturas que se pode colocar sobre uma pizza é quase infinita, entretanto, algumas preparações são tradicionais e têm fiéis seguidores:



Margherita



Mussarela



Portuguesa



Calabresa



Califórnia



Pepperoni



Quatro queijos



Bacon

Prática 17

- Altere a página pizza.html para que seu conteúdo esteja envolvido em uma tabela com:
 - 3 linhas
 - Alinhamento centralizado
 - Largura de 80%
 - Cellspacing e cellpadding de 10 pixels
 - Cor de fundo branca
- A imagem do topo deve ficar em uma célula (sugestão: mude a imagem para a imagem topo1.jpg)
- O conteúdo deve ficar na linha do meio
- A última linha deve conter o rodapé (www.pizzaonline.com.br)
- Coloque como fundo do body a imagem fundo_listras.jpg

Ilustração no próximo slide

Linha 1,
coluna 1
(cabeçalho)



topo1.jpg



Pizza

Pizza é uma preparação culinária que consiste em um disco de massa fermentada de farinha de trigo, regado com molho de tomates e coberto com ingredientes variados que normalmente incluem algum tipo de queijo, carnes preparadas ou defumadas e ervas, normalmente orégano ou manjerição, tudo assado em forno.



Alguns tipos e sabores

A variedade de coberturas que se pode colocar sobre uma pizza é quase infinita, entretanto, algumas preparações são tradicionais e têm fiéis seguidores:



Marqherita



Mussarela



Portuguesa



Calabresa



Califórnia



Pepperoni



Quatro queijos



Bacon



Fontes

1. <http://pt.wikipedia.org/wiki/Pizza>
2. <http://www.pizza.it>
3. http://en.wikipedia.org/wiki/History_of_pizza

www.pizzaonline.com.br

Linha 2,
coluna 1
(corpo)

fundo_listras.jpg

Linha 3,
coluna 1
(rodapé)

Tabelas

- Pode ser necessário expandir uma célula ocupando o local de outras
- Essa combinação, expansão ou mesclagem de células é obtida usando-se os atributos colspan e rowspan da tag `<td>`

Ex:

Conteúdo do Curso	
Aplicar recursos avançados de processador de texto e planilha eletrônica.	
Material instrucional indisponível.	
Disciplina	Carga Horária
Word Avançado	20 horas
Excel Avançado	20 horas
Carga Horária Total	40 horas

Conteúdo do Curso	
Aplicar recursos avançados de processador de texto e planilha eletrônica.	
Material instrucional indisponível.	
Disciplina	Carga Horária
Word Avançado	20 horas
Excel Avançado	20 horas
Carga Horária Total	40 horas

(1,1) e (1,2)	(1,3)
(2,1), (2,2) e (2,3)	

(1,1) e (2,1)	(1,2)	(1,3), (2,3) e (3,3)
	(2,2)	
(3,1)	(3,2)	

Tabelas

- colspan** (column span): faz com que uma célula ocupe o um número de células especificado. As células que forem ocupadas pela expansão devem ser retiradas.

(1,1)	(1,2)	(1,3)
(2,1)	(2,2)	(2,3)



(1,1) e (1,2)	(1,3)
(2,1), (2,2) e (2,3)	

```
<table border = "1">
  <tr>
    <td>(1,1)</td>
    <td>(1,2)</td>
    <td>(1,3)</td>
  </tr>
  <tr>
    <td>(2,1)</td>
    <td>(2,2)</td>
    <td>(2,3)</td>
  </tr>
</table>
```



```
<table border = "1">
  <tr>
    <td colspan = "2">
      (1,1) e (1,2)
    </td>
    <td>(1,3)</td>
  </tr>
  <tr>
    <td colspan = "3">
      (2,1), (2,2) e (2,3)
    </td>
  </tr>
</table>
```

Tabelas

- rowspan**: faz com que uma célula se expanda por linhas ao invés de colunas.

(1,1)	(1,2)	(1,3)
(2,1)	(2,2)	(2,3)
(3,1)	(2,2)	(2,3)



(1,1) e (2,1)	(1,2)	(1,3), (2,3) e (3,3)
	(2,2)	
(3,1)	(3,2)	

```
<table border = "1">
  <tr>
    <td>(1,1)</td>
    <td>(1,2)</td>
    <td>(1,3)</td>
  </tr>
  <tr>
    <td>(2,1)</td>
    <td>(2,2)</td>
    <td>(2,3)</td>
  </tr>
  <tr>
    <td>(3,1)</td>
    <td>(2,2)</td>
    <td>(2,3)</td>
  </tr>
</table>
```



```
<table border = "1">
  <tr>
    <td rowspan = "2">(1,1) e (2,1)</td>
    <td>(1,2)</td>
    <td rowspan = "3">
      (1,3), (2,3) e (3,3)
    </td>
  </tr>
  <tr>
    <td>(2,2)</td>
  </tr>
  <tr>
    <td>(3,1)</td>
    <td>(3,2)</td>
  </tr>
</table>
```

Tabelas


- Legenda para tabela
 - `<caption align="top"|"bottom">[</caption>]`
 - O atributo align define se o título vai aparecer na parte superior (top) ou na parte inferior (bottom) da tabela.
 - O texto delimitado por esta tag será exibido como título da tabela, centralizado e em negrito.
 - A tag caption deve aparecer imediatamente após a tag table e antes das linhas da tabela.

Prática 18

- Crie um arquivo chamado **automoveis.html** e nele crie uma tabela semelhante à tabela abaixo com células mescladas:

Vendas de Automóveis no Varejo por Modelo			
Fabricante	Marca (incluindo todos os modelos e categorias)	1995	1996
Fiat	Tempra	12.345	15.670
	Uno	101.275	105.692
Ford	Fiesta	54.888	63.505
General Motors	Corsa	84.163	85.115
	Monza	24.732	19.781
	Vectra	14.334	15.128
Volkswagem	Gol	122.300	129.782
	Parati	22.343	19.942

Formulários



To sign up for our regular newsletter containing exclusive offers, please enter your details below:

Email:

Postcode:

Please select any areas of special interest – You may choose as many categories as you wish:

- ☒ Baby and Toddler
- ☒ Preschool and Educational
- ☒ Toys and Games
- ☐ Bikes and Wheel Goods
- ☐ Sports and Outdoor
- ☐ Video Games
- ☐ Electronics

Your privacy is important to Toys "R" Us. Please view our [Privacy Policy](#) before you submit your information.

If you wish to unsubscribe to the marketing material, please enter your email address here:

Email:

Panel 1 "Testing panel"

Dimensions

Temperature (°C)

Connections

Name

Address

☒ Active

Status ☐ Idle ☒ Running

State ☐ Old ☒ New

Menu

Salad

Snails

Butter

Trout

Shopping List

This is only a testing panel

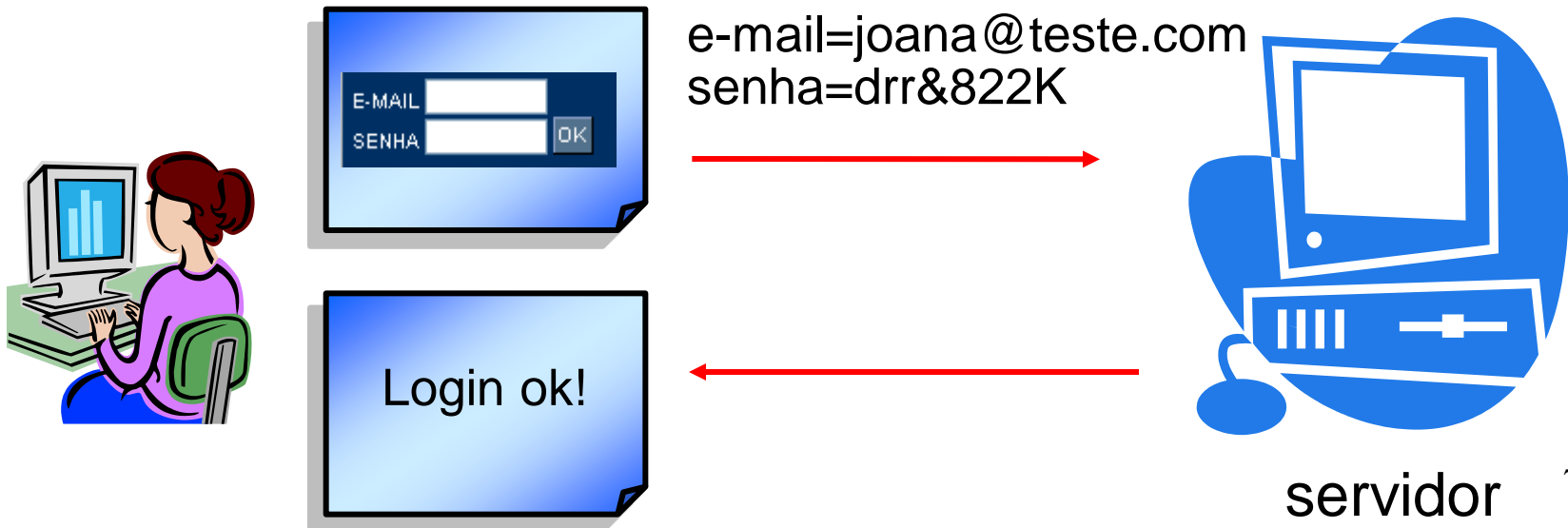
Web Shopping Imagens Notícias Dicionários

E-MAIL @uol.com.br

SENHA

Formulários

- Metáforas baseadas nos formulários de papel para preenchimento de inscrições, pedidos, cadastros
- São responsáveis pela interação entre um usuário e o servidor
- O resultado do preenchimento de um formulário é o envio dos dados para um servidor que suporte programação para a web dinâmica



Formulários

- Alguns campos/componentes de formulários são:

- Texto (text):

- Senha (password):

- Texto longo (textarea):

- Múltipla escolha (checkbox):

- Escolha única (radio):

- Caixas de seleção (select):

- Botões (buttons):

- Campos ocultos

Formulários

- Criando formulários

<html>

....

<form action = "http://..." method = "post">

A diagram representing a form with four horizontal input fields. Each field is a rectangle with a red border. To the left of each field is a red arrow pointing left, indicating the direction of input or a label.

Campos
ou componentes

</form>

....

</html>

Formulários

- **<form>**: tag dupla que deve “envolver” todos os campos. Principais atributos:
 - **action**: URL da página do script que executará a “ação”
 - Ex: `www.teste.com.br/cadastro.php`
 - **method**: especifica como os dados dos formulários são passados para a página de scripts. Valores:
 - `get` : dados são concatenados com a url.
Ex:
`www.teste.com.br/cadastro.php?nome=ely&cod=3rt55W`
 - `post`: dados são enviados no corpo do formulário na mensagem HTTP

Formulários - Texto e de senha

- **<input/>**: tag simples que define uma entrada de dados.

Principais atributos:

- **name**: nome da caixa de texto
 - **type**: identifica o tipo de entrada. Valores: **text** para texto simples e **password** para campos de senha
 - **size** : tamanho da caixa de texto
 - **maxlength**: número máximo de caracteres possíveis do campo
 - **value**: valor inicial (opcional)
 - **placeholder**: dica de conteúdo para o campo (opcional)
- Ex: (*tag form omitida*)

Nome: **<input type= “text” name = “txt_nome” size=“10” maxlength= “20” value= “Digite seu nome”/>
**

Senha: **<input type= “password” name = “pwd_senha” size=“10” maxlength= “10”/>**

Endereço: **<input type= “text” name = “txt_endereco” size=“40” maxlength= “40” placeholder=“Digite seu endereço”/>**

Formulários - Múltipla escolha

- **<input/>**: novamente uma tag input. Atributos:
 - **name**: nome do campo
 - **type**: deve ser **checkbox**
 - **value**: Valor do campo (obrigatório)
 - **checked**: define se o checkbox está marcada
- Ex: (*tag form omitida*)
<input type = “checkbox” name = “ckb_hobbies” value = “pescar” />Pescar **
**
...
<input type = “checkbox” name = “ckb_hobbies” value = “dancar” checked />Dançar

Formulários - Escolha única

- **<input/>**: novamente uma tag input. Atributos:


- **name**: nome do campo

- **type**: deve ser **radio**

- **value**: valor do campo (obrigatório)



- **checked**: define se a seleção está marcada

Marcando um,
o outro é desmarcado



- Ex: (*tag form omitida*)

Funcionam em conjunto
se tiverem o mesmo valor
para o atributo name



```
<input type= "radio" name= "rdb_sexo" value=
"m"/>Masculino<br/>
```

```
<input type= "radio" name= "rdb_sexo" value=
"f"/>Feminino<br/>
```

Formulários – Texto longo

- Texto longo ou com várias linhas
- **<textarea>**: Tag dupla. Atributos:
 - name: nome do campo
 - rows: número de linhas
 - cols: número de colunas (caracteres por linha)
- Ex: *(tag form omitida)*

<textarea cols="100" rows="5" name="txt_comentarios">

Digite aqui seu comentário

</textarea>

**<textarea cols="100" rows="5" name = "txt_comentarios"
placeholder="Digite aqui seu comentário"></textarea>**

Formulários - Caixas de seleção

- **<select>** : tag dupla que representa um contêiner de opções. Seu padrão é no formato combo-box. Atributos:
 - name: nome do campo
 - size: caso seja maior que 1 o componente será exibido em forma de lista
- **<option>** : tag dupla que representa cada item.
 - selected: define se a “seleção está selecionada”
 - value: o valor de cada option.

• Ex: *(tag form omitida)*

```
<select name = "slt_estado">  
  <option value = "ce"> Ceará </option>  
  <option value = "ma"> Maranhão </option>  
  <option value = "pi" selected> Piauí</option>  
</select>
```

Formulários – Botões de ação

- **<input>** novamente uma tag input. Atributos:
 - name: nome do campo
 - type: deve ser **submit** para enviar ou **reset** para restaurar o estado original dos campos do formulário
 - value: Texto que é exibido no botão
- Ex: (*tag form omitida*)

```
<input type= “submit” name = “btn_entrar” value = “Entrar”/>
```

```
<input type= “reset” name = “btn_limpar” value = “Limpar”/>
```


HTML 5 – Novos tipos de campo

Muitas necessidades que os desenvolvedores possuíam e só preenchiam através do javascript foram implementadas no HTML5. Em apenas algumas linhas é possível disponibilizar componentes como: slider, autocomplete, campo data, além de validações de diversos campos. Veja algumas destas opções para formulários:

Telefone – `<input name=tel type=tel>` - Utilizado para entrada de dados de telefones

Tel:

URL – `<input name=url type=url>` – Utilizado para entrada de variadas URLs. Por padrão o browser irá inserir o http:// como protocolo padrão

URL:

E-mail – `<input name=email type=email>` – Caso opte por validação, automaticamente o browser valida se o valor for um e-mail válido

E-mail:

HTML 5 – Novos tipos de campo

Data e hora – `<input name=horario type=datetime>` – Utilizado para agendamento de eventos, reuniões, etc.

Data/hora:

Número – `<input name=numero type=number>` – Com os atributos “min” e “Max” é possível entrar com um intervalo de valores possíveis e com o “step” é possível definir o valor para cada incremento

Número:

Tempo – `<input name=tempo type=time>`

Tempo:

Semana – `<input name=semana type=week>`

Semana:

Slider – `<input name=slider type=range min=2 max=30 step=2 >`

Slider

HTML 5

Atributo autofocus

Uma boa recomendação para melhorar a acessibilidade e usabilidade de uma página é já deixar o campo principal do cursor selecionado (focus no campo). Normalmente utilizamos a linha de javascript `(document.getElementById('nome_capo').focus())` para deixar o campo focado.

```
<input maxlength="256" name="q" value="" autofocus>
```

Validações de Formulários

Não sera mais necessário a utilização de javascript para validação de formulários, no HTML5 existe o atributo "required", que dentro de um elemento input torna o campo obrigatório e transfere para o browser a tarefa da validação do campo.

```
<p><label>Nome: <input name=name required></label></p>
```

```
<p><label>Tel: <input name=tel type=tel required></label></p>
```

```
<p><label>E-mail: <input name=email type=email required></label></p>
```

```
<p><label>URL: <input name=url type=url required></label></p>
```

```
<button>Enviar</button>
```

HTML 5

Validação por expressão regular

Além de deixar o campo apenas como obrigatório, é possível colocar um padrão de entrada para aquele determinado campo, que será validado pelo browser.

Por exemplo, em um campo em que só possa entrar valores numéricos com 3 dígitos:

```
<input pattern="[0-9]{3}"  
name="digito"  
required  
title="Validação apenas para 3 dígitos"/>
```

Observe que o atributo pattern aceita expressões regulares.

Validação por range de valores

No tipo de campo “numérico”, é possível fazer uma validação dos valores possíveis de entrada utilizando os atributos min e max.

```
<input type=number min=2 max=10 />
```

HTML 5

- **Autocomplete**

- Com HTML5 é simples exibir um input com “autocompletar”, este recurso é bem interessante, pois melhora significativamente a experiência do usuário no preenchimento do formulário – quando bem utilizado.

```
<label>
```

```
Homepage:<input name="hp" type="url" list="hpurls">
```

```
</label>
```

```
<datalist id="hpurls">
```

```
<option value="http://www.ifpi.edu.br/" label="IFPI">
```

```
<option value="http://www.terra.com.br/" label="TERRA">
```

```
<option value="http://www.g1.com.br/" label="G1">
```

```
</datalist>
```

- **TAG Fieldset**

```
<fieldset>
```

```
<legend align="tipo">texto</legend>
```

```
CAMPOS
```

```
</fieldset>
```



Parte 1

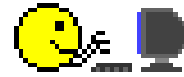
URL :

E-mail :

Formulários: id x name

- O atributo **name** dos campos de formulário está em desuso
- Atualmente usa-se um atributo **id** com valor igual ao do atributo **name** sob a recomendação HTML5
- O atributo **id** é muito utilizado como identificador único e é usado em outros WebStandards
- Por motivos didáticos, foi omitido o uso do **id**, mas quando estivermos utilizando JavaScript faremos um uso mais intenso desse atributo

Prática 19





- Crie uma página para cadastros de leitores do Pizza On Line conforme a ilustração no próximo slide e coloque um link para a mesma na página pizza.html
- Use o layout em tabelas e a figura de fundo da página fundo_listras.jpg
- Atente aos outros detalhes como cores e figuras e demais imagens
- Seja criativo, ouse!!!

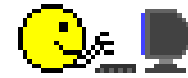
Ilustração no próximo slide

Prática 19


- Ilustração com bordas nas tabelas:

	
 Cadastre-se e receba nossas atualizações	
<hr/>	
Nome:	<input type="text"/>
Senha:	<input type="password"/>
Confirmação de senha:	<input type="password"/>
Seu tipo de Pizza preferido:	<input type="text" value="Margherita"/>
Quantas vezes come Pizza por semestre?	<input type="radio"/> Entre 1 a 5 <input type="radio"/> 5 a 10 <input type="radio"/> Mais de 10
Comentário:	<input type="text"/>
<input checked="" type="checkbox"/> Deseja receber e-mails dos nossos parceiros	
<input type="button" value="Enviar"/> <input type="button" value="Limpar"/>	
<hr/>	
www.pizzaonline.com.br	
	

Prática 19



- Ilustração sem bordas:

 **Cadastre-se e receba nossas atualizações**

Nome:

Senha:

Confirmação de senha:

Seu tipo de Pizza preferido: ▼

Quantas vezes come Pizza por semestre?

☐ Entre 1 a 5

☐ 5 a 10

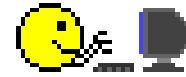
☐ Mais de 10

Comentário:

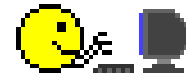
☒ Deseja receber e-mails dos nossos parceiros

www.pizzaonline.com.br

Prática 20



- Crie uma página chamada **busca.html** e nela crie um formulário para realizar uma busca no Google.
- Configuração do formulário(valores que o google exige):
 - `action` = "<http://www.google.com/search>"
 - `method` = "`get`"
 - Caixa de texto com o nome: "`q`"
 - Atributos do Input **Pesquisar na WEB**: `id`="`all`", `name`="`lr`"
 - Atributos do Input **Pesquisar Páginas em Português**: `id`="`il`", `name`="`lr`", `value`="`lang_pt`"
- Os demais detalhes podem ser personalizados
- *Ilustração no próximo slide*



Google

Termo:

Buscar

Selecione o idioma:

☒ Todos os idiomas ☐ Português ☐ Espanhol ☐ Italiano

Feito por xxxx

Caracteres de Escape

- Os caracteres "<" e ">", por definirem o início e final dos descritores, não podem ser impressos na tela.
- Quando é necessário exibi-los, deve-se usar uma seqüência de escape.
- Esta seqüência é iniciada por um "&" seguido de uma abreviação e um ponto-e-vírgula, que indica o final da seqüência.
- Como o "&" também é caractere especial, há também uma seqüência para exibi-lo.

Caracteres de Escape

<i>Caractere</i>	<i>Sequência de Escape</i>
<	<
>	>
&	&
"	"

- Para produzir o resultado:

144 < 25 + x < 36 + y

Fulano, Sicrano & Cia.

- É preciso ter o seguinte código HTML:

144 < 25 + x < 36 + y

Fulano, Sicrano & Cia.

Caracteres de Escape

- Também são usados para produzir caracteres especiais ou letras acentuadas:

Plantação

maßgebend

Håndbøger

Enciclopædia

©

sueño

- HTML:

Plantação

maßgebend

Håndbøger

Enciclopædia

©

sueño

Caracteres de Escape

Entidade	Caracter
á	á
â	â
à	à
ã	ã
ç	ç
é	é
ê	ê
í	í
ó	ó
ô	ô
õ	õ
ú	ú
ü	ü

Entidade	Caracter
Á	Á
Â	Â
À	À
Ã	Ã
Ç	Ç
É	É
Ê	Ê
Í	Í
Ó	Ó
Ô	Ô
Õ	Õ
Ú	Ú
Ü	Ü

Lista completa disponível em:

<http://erikasarti.com/html/acentuacao-caracteres-especiais/>

Elementos do head

- Contém informações sobre o conteúdo do documento para fins de indexação e organização.
- Não contém informação que será exibida na página.
- **Title**
 - `<title>` é o único elemento obrigatório do bloco do cabeçalho.
 - Deve conter o título do documento que aparece fora da página, na barra de título do browser.
 - É o que aparece também nos **favoritos**.
 - O título deve conter informações que descrevam o documento.
`<title>HTML e CSS: Introdução</title>`

Elementos do head

- meta

- `<meta>` é usado para incluir meta-informação como palavras-chave, descrições, etc. que podem ser usadas por mecanismos de busca, softwares de pesquisa e catalogação.
- A informação adicional deve vir nos atributos:
 - `name` - descreve o tipo de meta-informação, por exemplo Keywords.
 - `content` - descreve o conteúdo da meta-informação, por exemplo, uma lista de palavras-chave separadas por vírgula.
 - `http-equiv` - usado para adicionar ou redefinir cabeçalhos HTTP. Neste caso, o `content` deve conter o conteúdo do cabeçalho.
- Exemplos:

```
<meta http-equiv="set-cookie" content="pag=12"/>
```

```
<meta name="keywords" content="html, css, folhas de  
estilo, estilo"/>
```

```
<meta name="description" content="esta página explica os  
fundamentos básicos de html e folhas de estilo usando a  
linguagem css."/>
```

```
<meta http-equiv="refresh"  
content="10;url=pagina.html"/>
```

```
<meta http-equiv="content-type" content="text/html;  
charset=iso-8859-1"/>
```

Elementos do head

- **Link e Script:**

- São usadas para vincular uma página a outro recurso.

Ex1: Ligando a um arquivo CSS

```
<link rel = "stylesheet"  
      type = "text/css"  
      href = "estilos/default.css"/>
```

Ex2: Ligando a um arquivo JavaScript

```
<script type = "text/javascript"  
      src = "alomundo.js">  
</script>
```

Elementos do head

- **BASE**

- `<base/>` altera os vínculos de origem e destino da janela.
- Normalmente a URL base de origem é o local onde a página se encontra (protocolo atual, máquina atual e diretório atual).
- Com `<base/>`, pode-se alterar a URL base para que as imagens e links sejam buscados em outros lugares.
- Normalmente a janela atual (`_self`) é a responsável por receber o resultado dos vínculos.
 - Para fazer com que links abram em outra janela usa-se `_blank`.
- Exemplos:

```
<base href = "." target = "_blank"/>
```

```
<base href = "http://www.a.com/dados/">
```

```
<!--os links relativos serão procurados em  
http://www.a.com/dados/-->
```

```
<base target = "lateral2">
```

```
<!-- links abrirão na janela lateral2 -->
```

HTML 5

- **Elementos de estrutura**
- **<header>** - cabeçalho da página ou de uma seção (não confundir com a tag <head>);
- **<nav>** - o conjunto de links que formam a navegação, seja o menu principal do site ou links relacionados ao conteúdo da página;
- **<section>** - cada seção do conteúdo;
- **<article>** - um item do conteúdo dentro da página ou da seção;
- **<aside>** - conteúdo relacionado ao artigo (como arquivos e posts relacionados em um blog, por exemplo).
- **<footer>** - o rodapé da página ou de uma seção;

HTML 5



HTML 5

- **Elementos de conteúdo**

<figure> - usado para associar uma legenda a uma imagem, vídeo, arquivo de áudio, objeto ou iframe:

<canvas> - através de uma API gráfica, irá renderizar imagens 2D dinâmicas que poderão ser usadas em jogos, gráficos, etc;

<dialog> - junto com as tags **<dt>** e **<dd>** será usado para formatar um diálogo;

<meter> - usada para representar medidas, que podem ser de distância, de armazenagem em disco etc.;

```
<figure id="figura01">  
<legend>Figura 1. Esquema de uma página em HTML 5</legend>  
  
</figure>
```

HTML 5

- **dialog**, que serve para indicar um diálogo, como abaixo:

<dialog>

<dt> Costello

<dd> Look, you gotta first baseman?

<dt> Abbott

<dd> Certainly.

<dt> Costello

<dd> Who's playing first?

<dt> Abbott

<dd> That's right.

<dt> Costello

<dd> When you pay off the first baseman every month, who gets the money?

<dt> Abbott

<dd> Every dollar of it.

</dialog>

HTML 5

- **nav**, que serve para indicar uma área com vários links, como por exemplo uma nuvem de tags num blog;

```
<nav>
```

```
<ul>
```

```
<li> <a href="/"> Home </a> </li>
```

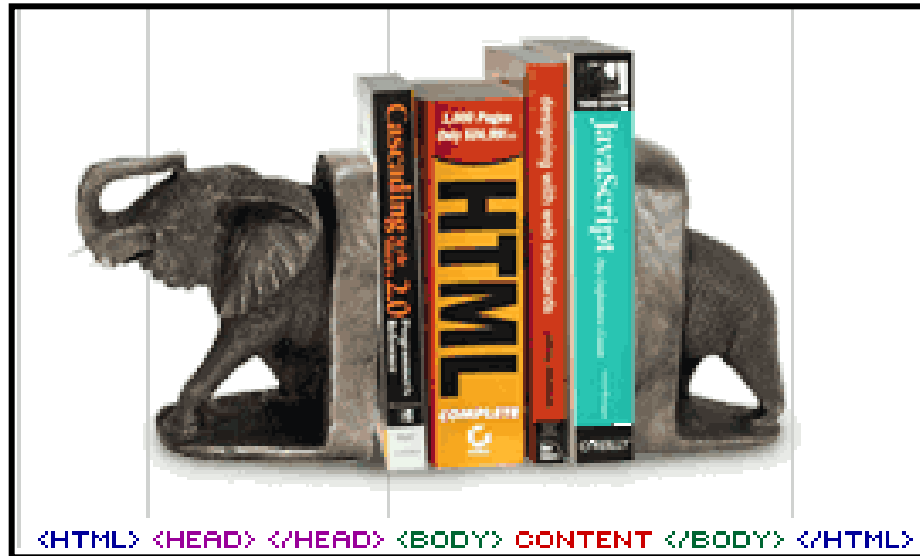
```
<li> <a href="/products"> Products </a> </li>
```

```
<li><a href="/services"> Services </a></li>
```

```
<li><a href="/about"> About </a></li>
```

```
</ul>
```

```
</nav>
```

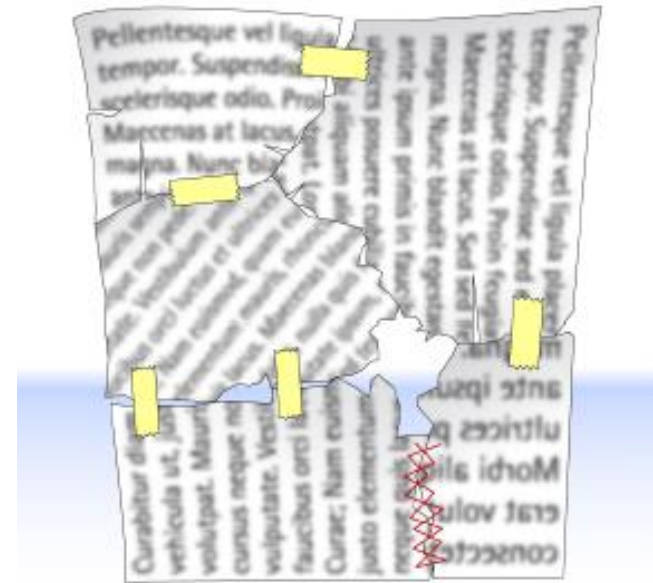
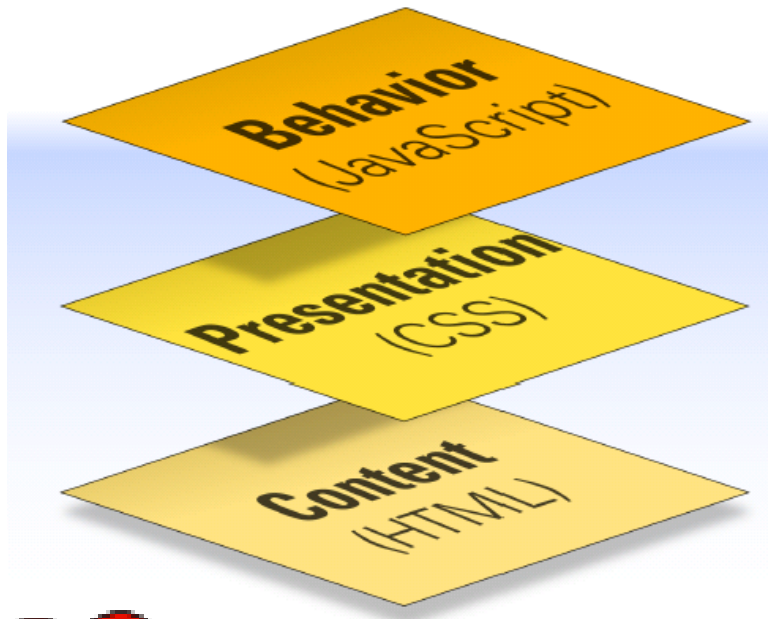
Programação Web Front-End

Introdução ao CSS

Estrutura de páginas

- O código em uma página pode ser concebido em três visões/camadas distintas:
 - Estrutura e conteúdo: **HTML**
 - Formatação/Apresentação: **CSS**
 - Comportamento: **JavaScript**

Estrutura de páginas



CSS

- Cascading Style Sheets (folhas de estilos em cascata)
- Padronizado pelo W3C (World Wide Web Consortium)
- Permitem formatação “fora” do HTML: muda-se o estilo da página, porém o HTML fica o mesmo
- Principais vantagens:
 - páginas mais flexíveis
 - mais leves
 - maior controle do seu aspecto gráfico com maior precisão
 - melhor manutenção
 - maior facilidade na correção de erros

Inserindo CSS no HTML

- Arquivo externo usando a tag <link>

```
<link rel="stylesheet" type="text/css" href="teste.css"/>
```

- Dentro da tag <style> da seção <head> do HTML

```
<style>
```

```
h1 {  
    color: green;  
    font-size: 12px;  
    text-align: center;  
    font-family: verdana;  
}
```

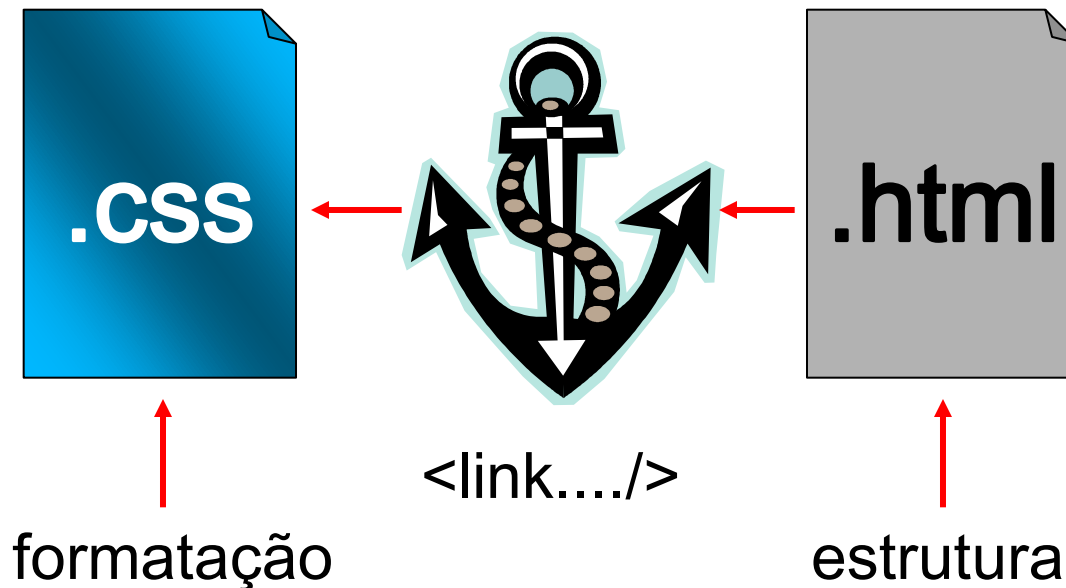
```
</style>
```

- Dentro da tag HTML

```
<h1 style="color:green; font-size:12px">Texto</h1>
```

Dois arquivos separados

- Usaremos dois arquivos texto:
 - Um com HTML com extensão .html
 - Outro com CSS com extensão .css
- Haverá ainda uma tag HTML que “unirá” os arquivos



Juntando HTML + CSS

- **<link/>** : tag simples que fica dentro da seção **<head></head>** que “unirá” um arquivo CSS a um arquivo HTML. Atributos:
 - **rel** : diz a que conteúdo a tag é relativa. No nosso caso será sempre o valor **stylesheet**
 - **type** : define o tipo de conteúdo. No nosso caso será sempre o valor **text/css**
 - **href** : define a URL onde está o CSS. A grosso modo é o nome do arquivo CSS. Por padrão, o nome do arquivo CSS vai ser o mesmo do arquivo HTML.
Ex: Página **teste.html** → **teste.css**
 - **media** : dispositivo que será exibido: all | aural | braille | handheld | projection | print | screen | tty | tv |

Juntando HTML + CSS

- Ligação com o arquivo HTML através de uma tag `<link/>` supondo que uma página com o nome **teste.html**:

...

`<head>`

`<title>teste de uso de css</title>`

`<link rel="stylesheet" type="text/css"`
`href="teste.css" media="all"/>`

`</head>`

...

Sintaxe CSS

- A sintaxe das definições CSS é composta por três partes: seletor, uma declaração e valor
- Sintaxe básica: **seletor { propriedade: valor; }**
 - **Seletor:** inicialmente, é o elemento HTML identificado por sua tag. Posteriormente veremos que o seletor pode ser uma classe ou um ID
 - **Propriedade:** é um atributo do elemento HTML ao qual será aplicada a regra (ex: font, color, background, etc...)
 - Nota: dificilmente o atributo em CSS terá o mesmo nome do atributo no HTML
 - **Valor:** é a formatação específica a da propriedade (ex: fonte arial, cor azul, alinhamento ao centro, etc...)

Ex: formatar as tags **<h1>** com cor verde, alinhadas ao centro, fonte de 12 pixels e família verdana

teste.css



```
h1 {color: green;}  
h1 {font-size: 12px;}  
h1 {text-align: center;}  
h1 {font-family: verdana;}
```

Sintaxe CSS

- Escreve-se o seletor e, entre chaves { }, a propriedade e valor separados por dois pontos e entre chaves.
- Quando mais de uma propriedade for definida na regra, podemos compactá-la usando ponto-e-vírgula para separá-las.

```
h1 {color: green;}  
h1 {font-size: 12px;}  
h1 {text-align: center;}  
h1 {font-family: verdana;}
```



```
h1 {  
    color: green;  
    font-size: 12px ;  
    text-align: center;  
    font-family: verdana;  
}
```

- Um detalhe importante é que, existindo várias tags **<h1>**, a formatação já estará definida a cada uma delas
- Comentários: definidos entre **/* comentário */**

Prática 21

- Altere o arquivo `exame.html` para usar CSS tendo com características:
 - Todas as tags `<h1></h1>` tenham fonte arial
 - O título esteja centralizado
 - As tags `<h3></h3>` até `<h6></h6>` tenha uma cor distinta

Exame de vista

Leia apenas o que se pede:

números : a 2 3 u 5 j 8

letras : y a 1 2 / 3 q

vogais : 1 a + o p 3 i

consoantes : c a 8 o # d i

Definindo classes

- Podemos definir classes diferentes para um determinado elemento/seletor HTML
- Podemos criar um nome e com ele criar uma **classe** a qual definirá as regras CSS

Ex:

p.centralizado {font-size: 20px; text-align: center; font-family: arial;}

p.direita {font-size: 12px; text-align: right; font-family: verdana;}

← CSS

<p class="centralizado">Curriculum Vitae</p>

<p class="direita" >Informações pessoais</p>

← HTML

- Nesse caso, o atributo **class** das tags HTML é uma forma de definir qual das formatações/classes definidas no arquivo CSS deve ser aplicada ao parágrafo

Classes sem descritores

- Uma classe também pode não pertencer a uma tag específica:

```
.verde { color: green; }
```

- Todos os descritores que tiverem o atributo **class="verde"** serão formatados da mesma maneira:

```
<p class="verde">...</p>  
<h3 class="verde">...</h3>  
etc.
```

- Com essa abordagem, tem-se a vantagem de reusar definições em diferentes tags

Agrupando Seletores

- Quando os seletores têm as mesmas propriedades, podemos agrupar as definições:

p { font-family: arial; }

h1 { font-family: arial; }

h2 { font-family: arial; }

h3 { font-family: arial; }  p,h1,h2,h3,h4,h5,h6 { font-family: arial; }

h4 { font-family: arial; }

h5 { font-family: arial; }

h6 { font-family: arial; }

Herança com tags aninhadas

- Quando uma tag superior possui uma formatação CSS, as tags de “dentro” herdam as definições:

```
body { font-family: arial; color: blue;}
```

```
p { font-size : 12px;}
```

```
h1 { text-align: center;}
```



p e h1 herdam a fonte e a cor de **body**, pois estão aninhadas a ela

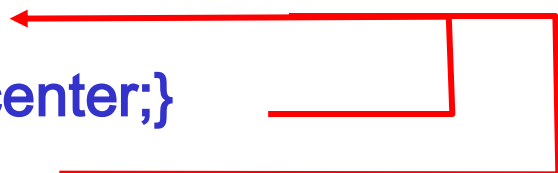


```
<body>  
  <p>...</p>  
  <h1>...</h1>  
</body>
```

Estilos em “cascata”

- Podemos definir estilos em cascata, definindo herança:

```
p {font-family: verdana; color: green;}  
p.centralizado {font-size: 20px; text-align: center;}  
p.direita {font-size: 12px; text-align: right;}
```

A red line starts from the right side of the first line of CSS code, goes right, then down, then left, ending with an arrow pointing to the 'p' in 'p.centralizado'. Another red line starts from the right side of the third line of CSS code, goes right, then down, then left, ending with an arrow pointing to the 'p' in 'p.direita'.

- Além da formatação própria, **p.centralizado** e **p.direita** “herdam” a formatação de **p**.
- No entanto, **p não herda** as definições dos outros dois

Seletor ID

- Um seletor ID só deve ser aplicado a uma única tag HTML dentro do documento
- Seu uso assemelha-se ao uso de classes
- Sintaxe: **#nomeID { propriedade: valor; }**
- Ex:

#titulo {font-size: 20px; text-align: center; font-family: arial} ← CSS
#subtitulo {font-size: 10px; font-family: verdana}

<h1 id = "titulo">Curriculum Vitae</h1> ← HTML
<h2 id = "subtitulo" >Informações pessoais</h2>

- Notas:
 - Idealmente, prefira usar nomes de IDs e classes apenas com letras
 - IDs são mais usados como identificadores de layout
 - Aprenderemos mais sobre IDs posteriormente

Valores para cores

- Há várias formas de se definir o uso de cores:

h1 {color: red}

h1 {color: #ff0000}

h1 {color: rgba(100%, 0%, 0%, 0.0)}

h1 {color: rgba(255, 0, 0, 0.5)}

FUNÇÃO rgba(red, green, blue, alpha).

O parâmetro alpha é um número entre 0.0 (completamente transparente) e 1.0 (completamente opaco)

Nota: “rgba” não deve conter espaços

Link com a listagem de cores no site do W3C:

http://www.w3schools.com/css/css3_colors.asp

Prática 22

- Crie uma página chamada `image.html` e use CSS para deixá-la conforme a figura abaixo e as instruções do próximo slide:



The diagram illustrates the layout of a web page for the song 'Imagine'. It features a title, a line of text, two paragraphs of lyrics, and a copyright notice, all separated by horizontal lines. Red arrows and numbers point to specific elements:

- 1** points to the title **Imagine**.
- 2** points to the text **Letra: John Lennon**.
- 3** points to the first paragraph of lyrics: **Imagine there's no Heaven
It's easy if you try
No hell below us
Above us only sky
Imagine all the people
Living for today**.
- 4** points to the second paragraph of lyrics: **Imagine there's no countries
It isn't hard to do
Nothing to kill or die for
And no religion too
Imagine all the people
Living life in peace**.
- 5** points to the third paragraph of lyrics: **You may say that I'm a dreamer
But I'm not the only one
I hope someday you'll join us
And the world will be as one**.
- 6** points to the copyright notice: **Copyright: Barbatana's Records**.

Prática 22

- 1: h2 com fonte arial, cor maroon e tamanho padrão (ou seja, não modifique)
- 2, 3, 4, 5 e 6: são parágrafos
2. e 3. possuem a mesma formatação: fonte arial, cor maroon e tamanho 12px
4. e 6. possuem formatação dos itens 2. e 3. e alinhamento centralizado
5. Mesma formatação dos itens 2. e 3. e alinhamento à direita

Note que todos os itens tem a mesma cor de fonte e tipo de fonte (arial) e alguns possuem poucas diferenças...



Prática 23



Nossas bebidas tropicais

1

Chá verde refrescante



Cheio de vitaminas e minerais, este elixir combina os grandes benefícios à saúde do chá verde com uma mistura de folhas de camomila e raiz de ginseng.

Suco de Framboesa concentrado



Combinando suco de framboesa com raspas de limão e essência cítrica, este suco gelado fará você se sentir aliviado e refrescado.

Elixir de uva



Uvas azuis e a essência da cereja misturada em uma base do chá do eucalipto é capaz de lhe deixar desestressado em questão de minutos.

Anti-oxidante de cereja e acerola



Deguste do sabor da cereja e da acerola juntos nesta híbrida bebida rica em vitamina C.

2

4

5

Não trabalhamos com bebidas alcoólicas.

Prática 23

1: h3 com fonte arial e cor #007e7e

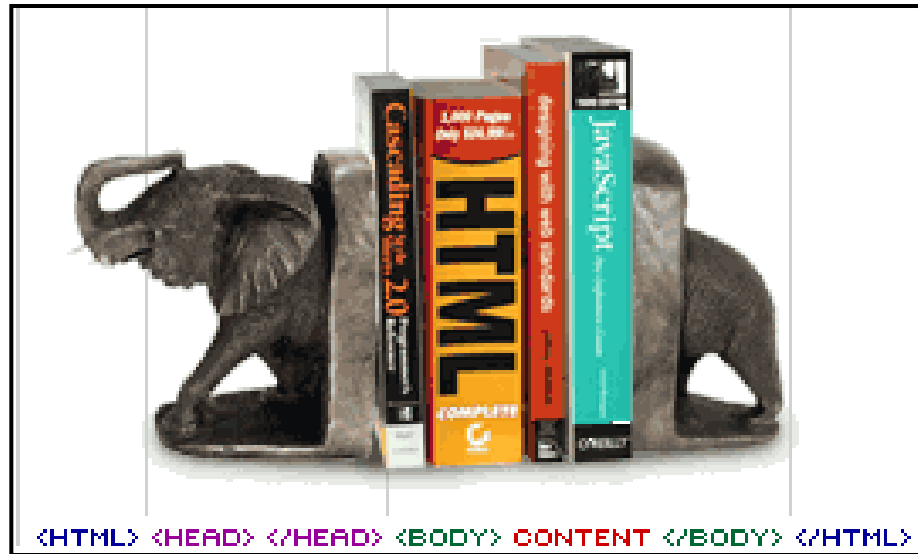
2: tags hr

3: h4 com fonte arial e cor #007e7e

4: parágrafos com fonte arial e cores: verde, azul, roxo (purple) e marron (maroon)

5: parágrafo com cor cinza, fonte arial e centralizado

Nota: procure sempre simplificar o CSS usando classes, cascata...



Programação Web Front-End

CSS: Tag Span, Formatação de textos, listas e fundos

Span

- **...** : é uma tag “neutra” e que não adiciona qualquer tipo de semântica ao documento
 - deve ser usada com blocos de texto quando não for necessária a quebra de linha
 - é considerada “inline” pois não possui quebras de linhas.
 - pode usada dentro de outras tags para obter efeitos específicos.
- Ex:
Esta frase se contradiz...não, na verdade, ela não o faz...



```
<p class = "azul">Esta frase se contradiz...  
  <span class = "verde">não, na verdade, ela não o faz...</span>  
</p>
```


Prática 24

- Crie o logotipo do Google usando formatação de cores em CSS e usando a tag span:
 - Tipo: Arial
 - Tamanho: 48px
 - Cores: variadas

The image shows the Google logo, which consists of the word "Google" in a sans-serif font. Each letter is a different color: 'G' is blue, the first 'o' is red, the second 'o' is yellow, 'g' is blue, 'l' is green, and 'e' is red.

Nota: tente fazer com outra tag, como parágrafo e veja que temos o inconveniente da quebra de linha

Formatação de textos e fontes

- **font-size**: já conhecido
- **color**: já conhecido
- **text-align**: **left** | **center** | **right** | **justify**
 - Define o alinhamento do texto
- **text-indent**: **número** | **porcentagem**
 - Quando usado em parágrafos, realiza a indentação da primeira linha pra direita.
- **font-style**: **italic**
 - Texto exibido em itálico
- **font-weight**: **bold**
 - Texto exibido em negrito

Texto justificado



Formatação de textos e fontes

- **text-transform: capitalize | uppercase | lowercase**
 - **capitalize**: coloca em maiúsculo a 1ª letra de cada palavra
 - **uppercase**: todas as letras em maiúsculo
 - **lowercase**: todas as letras em minúsculo
- **text-decoration: underline | overline | blink | line-through**
 - **underline**: texto sublinhado
 - **overline**: linha sobre o texto (sobrelinhado?)
 - **line-through**: texto tachado

Formatação de textos e fontes

- Ex: Arquivo CSS

```
.italico { font-style : italic }  
.negrito { font-weight : bold}  
.capitulado { text-transform : capitalize }  
.caixa_alta { text-transform : uppercase}  
.caixa_baixa { text-transform : lowercase }  
.sublinhado { text-decoration: underline}  
.sobrelinhado { text-decoration: overline}  
.tachado { text-decoration: line-through}  
.piscando { text-decoration: blink}  
.edentado {text-indent: 10%}
```

Formatação de textos e fontes

- Ex: Arquivo HTML

```
<p class = "italico">italico</p>
```

```
<p class = "negrito">negrito</p>
```

```
<p class = "capitulado">texto capitulado</p>
```

```
<p class = "caixa_alta">caixa alta</p>
```

```
<p class = "caixa_baixa">Caixa Baixa</p>
```

```
<p class = "sublinhado">texto sublinhado</p>
```

```
<p class = "sobrelinhado">texto sobrelinhado</p>
```

```
<p class = "tachado">linethrough</p>
```

```
<p class = "piscando">piscando</p>
```

```
<p class = "endented">texto endented</p>
```

Prática 25

- Altere a página imagine.html para que fique semelhante à figura abaixo:

Imagine

Letra: John Lennon

Imagine there's no Heaven
It's easy if you try
No hell below us
Above us only sky
Imagine all the people
Living for today

Imagine there's no countries
It isn't hard to do
Nothing to kill or die for
And no religion too
Imagine all the people
Living life in peace

You may say that I'm a dreamer
But I'm not the only one
I hope someday you'll join us
And the world will be as one

Copyright: Barbatana's Records

Além da formatação já existente:

1. Negrito
2. Negrito e itálico

Formatação de listas

- **list-style-type: disc | circle | square | decimal | lower-roman | upper-roman | lower-alpha | upper-alpha**
 - **disc, circle, square:** formatos de disco, círculo e quadrado para indicar os itens
 - **decimal:** numeração arábica
 - **lower-roman:** numeração romana em caixa baixa
 - **upper-roman:** numeração romana em caixa alta
 - **lower-alpha:** ordenação alfabética em caixa baixa
 - **upper-alpha:** ordenação alfabética em caixa alta

Listas com imagens

- **list-style-image:** url(url_da_imagem)
 - Coloca uma imagem como marcador da lista
 - Ex:
Supondo que a imagem seta_azul.gif esteja em um diretório relativo imagens

list-style-image: url(imagens/seta_azul.gif)

Listas não ordenadas:

- ▶ ICC
- ▶ IPW
- ▶ SO

Listas

```
/* todas as tags de texto definidas como: cor azulada,  
   família da fonte verdana e tamanho 12px */  
body {  
    color: #333388;  
    font-family: verdana;  
    font-size: 12px;  
}  
  
/* definições pra listas não ordenadas */  
ul.disco { list-style-type: disc; }  
ul.circulo { list-style-type: circle; }  
ul.quadrado { list-style-type: square; }  
ul.imagem { list-style-image: url(imagens/seta_azul.gif); }  
  
/* definições pra listas ordenadas */  
ol.decimal { list-style-type: decimal; }  
ol.romana { list-style-type: lower-roman; }  
ol.alfabetica { list-style-type: lower-alpha; }
```

Listas

Demonstração do uso de listas com HTML e CSS

Listas não ordenadas:

- ICC
- IPW
- SO

- ICC
- IPW
- SO

- ICC
- IPW
- SO

- ◉ ICC
- ◉ IPW
- ◉ SO

Listas ordenadas:

1. ICC
2. IPW
3. SO

- i. ICC
- ii. IPW
- iii. SO

- a. ICC
- b. IPW
- c. SO

Formatação de fundos

- **background-color**: define a cor de fundo de uma tag
 - Valores: nome da cor | #cor_hexadecimal | rgb(%, %, %) | rgb(0-255, 0-255, 0-255)

Variação de 0 a 255



- Ex. no css:

```
.fundo_azul {background-color: dodgerblue;}  
.fundo_cinza {background-color: #808080;}  
.fundo_vermelho {background-color: rgb(100%, 0%, 0%);}  
.fundo_teal {background-color: rgb(0, 128, 128);}
```

- Ex. no HTML:

```
...  
<body class="fundo_azul">  
  <p class="fundo_vermelho">Fundo vermelho</p>  
  <h4 class="fundo_cinza">Fundo cinza</h4>  
  <i class="fundo_teal">Fundo teal</i>  
</body>  
...
```

Formatação de fundos

- **background-image**: define imagens de fundo para um elemento
 - Valor: url("URL_da_imagem")
- **background-repeat**: define como uma imagem carregada vai se repetir. Valores:
 - **repeat-x**: repete a imagem nos eixos x
 - **repeat-y**: repete a imagem nos eixos y
 - **repeat**: repete a imagem nos eixos x e y (padrão)
 - **no-repeat**: não repete a imagem
- **background-attachment**: define se o fundo irá ou não se mover com o texto ou ficar fixo na tela
 - Valores: fixed e scroll

Formatação de fundos

- **background-position**: define a posição da imagem de fundo
- Os valores são fornecidos aos pares com para x e y separados por espaço
- Valores:
 - **background-position**: *porc_horizontal% porc_vertical%*
background-position: *comprimento comprimento*
 - **background-position**: *posição_horizontal posição_vertical*
posição_horizontal : left, center, right
posição_vertical: top, bottom, center
- **Ex: background-position**: 50% 10%
background-position: 200px 300px
background-position: center top

Formatação de fundos

- **Nota sobre fundos:**

- O Firefox possui uma implementação para fundos que necessita do seguinte trecho de código para que as constantes (left, center, right, top, bottom) funcionem adequadamente:

```
body {
```

```
padding: 0;
```

```
margin: 0;
```

```
height: 100%;
```

```
}
```

← Margem interna

← Margem externa

← Altura

Prática 26



Nota: toda a página possui fonte do tipo arial

1. Imagem simples com a figura ff.jpg (nada no CSS, apenas HTML)
2. Fundo com a figura fundo_final.jpg
3. h1 com as formatações: alinhado à direita, e cor de fundo #ccbb99
4. Parágrafo: cor do texto ccbb99, alinhamento justificado, negrito e endentação de 5%
5. h6 com as formatações: fonte em itálico e cor de fundo #ccbb99

Prática 26

Final Fantasy: The Spirits Within

Final Fantasy: The Spirits Within conta a história de uma Terra infestada de alienígenas no ano de 2065. Os humanos vivem em "cidades barreira", todos numa tentativa de livrar o planeta dos Phantoms (que significa Fantasma), uma misteriosa raça alienígena. A única esperança vem da cientista Aki Ross e de seu mentor, Dr. Sid, que tem um plano de destruir os Phantoms sem causar danos ao planeta, mas um general chamado Hein está determinado à usar o canhão espacial "Zeus" para destruir os Phantoms - mesmo que isso cause danos à Terra no processo.

Copyright 2007 SQUARE ENIX CO., LTD. All Rights Reserved.

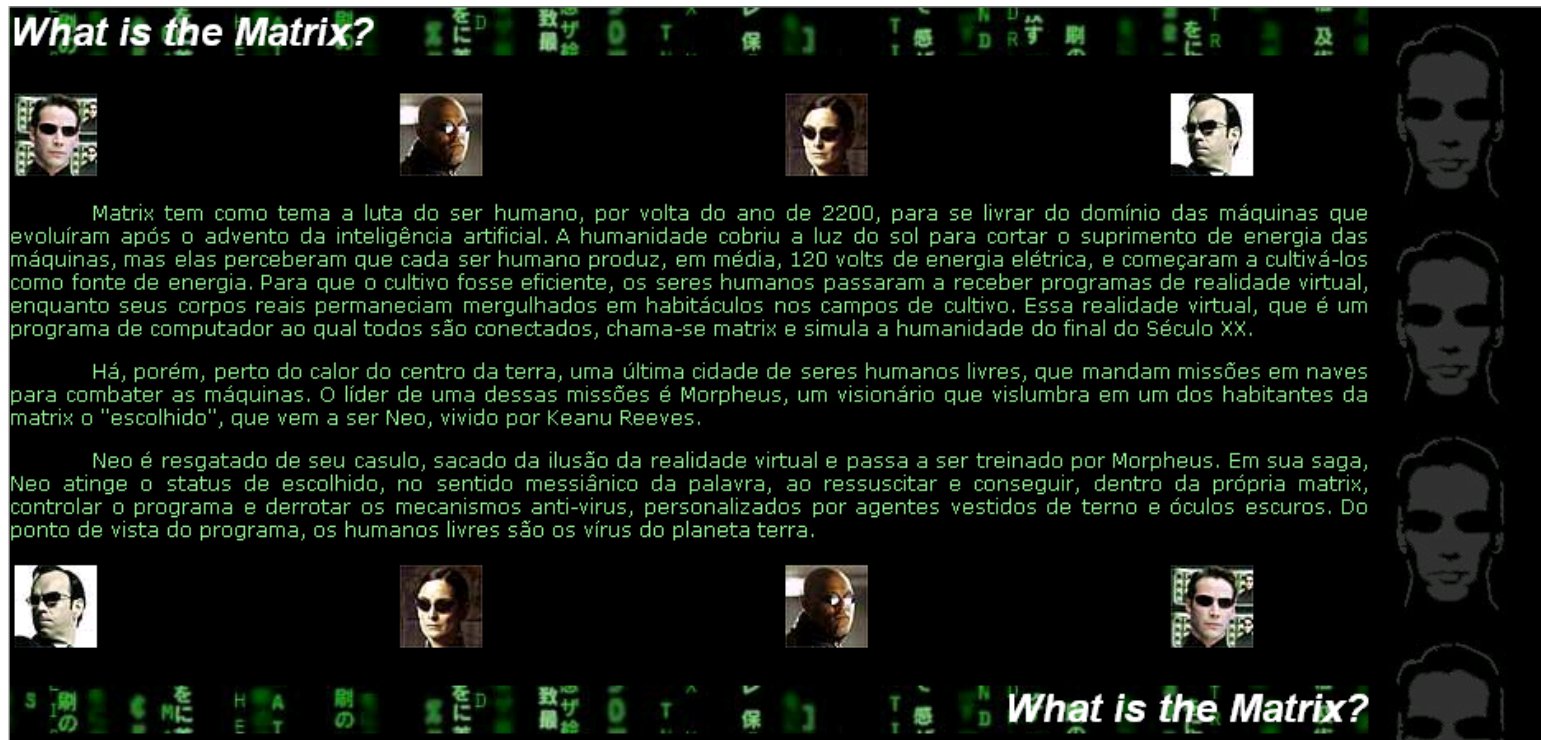
Prática 27



Além da formatação já existente:

1. h2 com as formatações: arial, cor do fundo marrom e cor da fonte #edc989
- 2: fundo com a figura fundo_lennon.gif
- 3: parágrafo com alinhamento centralizado, cor da fonte #edc989 e cor do fundo marrom

Prática 28



1. Corpo da página:
 - fundo preto;
 - imagem de fundo: fundo_neo.jpg;
 - repetição do fundo: somente no eixo y;
 - posição do fundo: na horizontal temos alinhamento à direita, na vertical pode ficar vazio;
2. Tabelas com as 4 fotos: largura de 100% (a tabela de baixo tem as imagens em outras posições);

Prática 28

3. h2 contendo o texto “What is the Matrix?”:

- fonte arial, cor branca e em itálico;
- fundo com a figura fundo_matrix.jpg;
- largura de 88%;
- o h2 da parte de baixo possui alinhamento à direita;

4 Parágrafos com o texto:

- fonte verdana, tamanho 12px e cor lightgreen;
- endentação de 50px;
- alinhamento justificado;
- largura de 88%;
- Nota: use a declaração width no CSS para dar a largura especificada nas construções dos parágrafos, tabela e h2 acima:

Ex:

```
h6 {... ; width = “88%”;    ... ; }
```

Prática 28

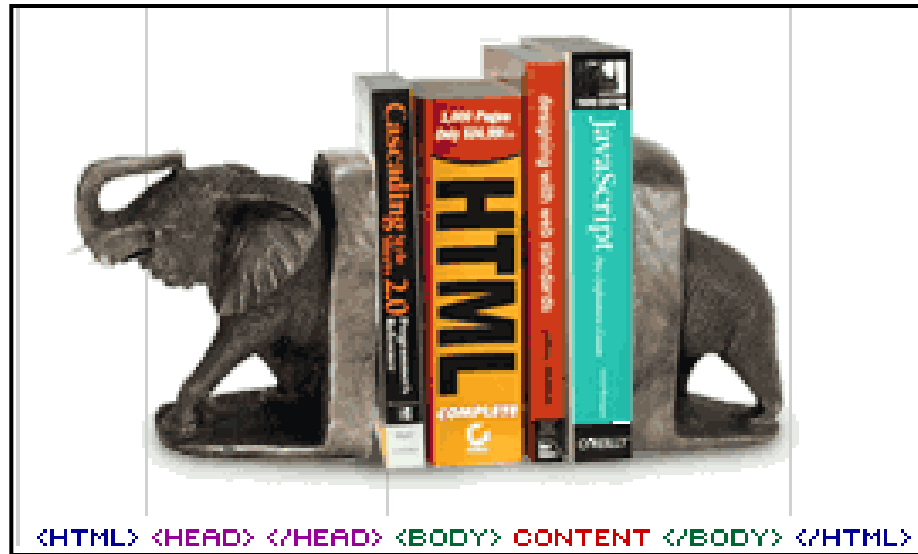
What is the Matrix?

Matrix tem como tema a luta do ser humano, por volta do ano de 2200, para se livrar do domínio das máquinas que evoluíram após o advento da inteligência artificial. A humanidade cobriu a luz do sol para cortar o suprimento de energia das máquinas, mas elas perceberam que cada ser humano produz, em média, 120 volts de energia elétrica, e começaram a cultivá-los como fonte de energia. Para que o cultivo fosse eficiente, os seres humanos passaram a receber programas de realidade virtual, enquanto seus corpos reais permaneciam mergulhados em habitáculos nos campos de cultivo. Essa realidade virtual, que é um programa de computador ao qual todos são conectados, chama-se matrix e simula a humanidade do final do Século XX.

Há, porém, perto do calor do centro da terra, uma última cidade de seres humanos livres, que mandam missões em naves para combater as máquinas. O líder de uma dessas missões é Morpheus, um visionário que vislumbra em um dos habitantes da matrix o "escolhido", que vem a ser Neo, vivido por Keanu Reeves.

Neo é resgatado de seu casulo, sacado da ilusão da realidade virtual e passa a ser treinado por Morpheus. Em sua saga, Neo atinge o status de escolhido, no sentido messiânico da palavra, ao ressuscitar e conseguir, dentro da própria matrix, controlar o programa e derrotar os mecanismos anti-vírus, personalizados por agentes vestidos de terno e óculos escuros. Do ponto de vista do programa, os humanos livres são os vírus do planeta terra.

What is the Matrix?



Programação Web Front-End

CSS: Links, tags div, box, model e layout

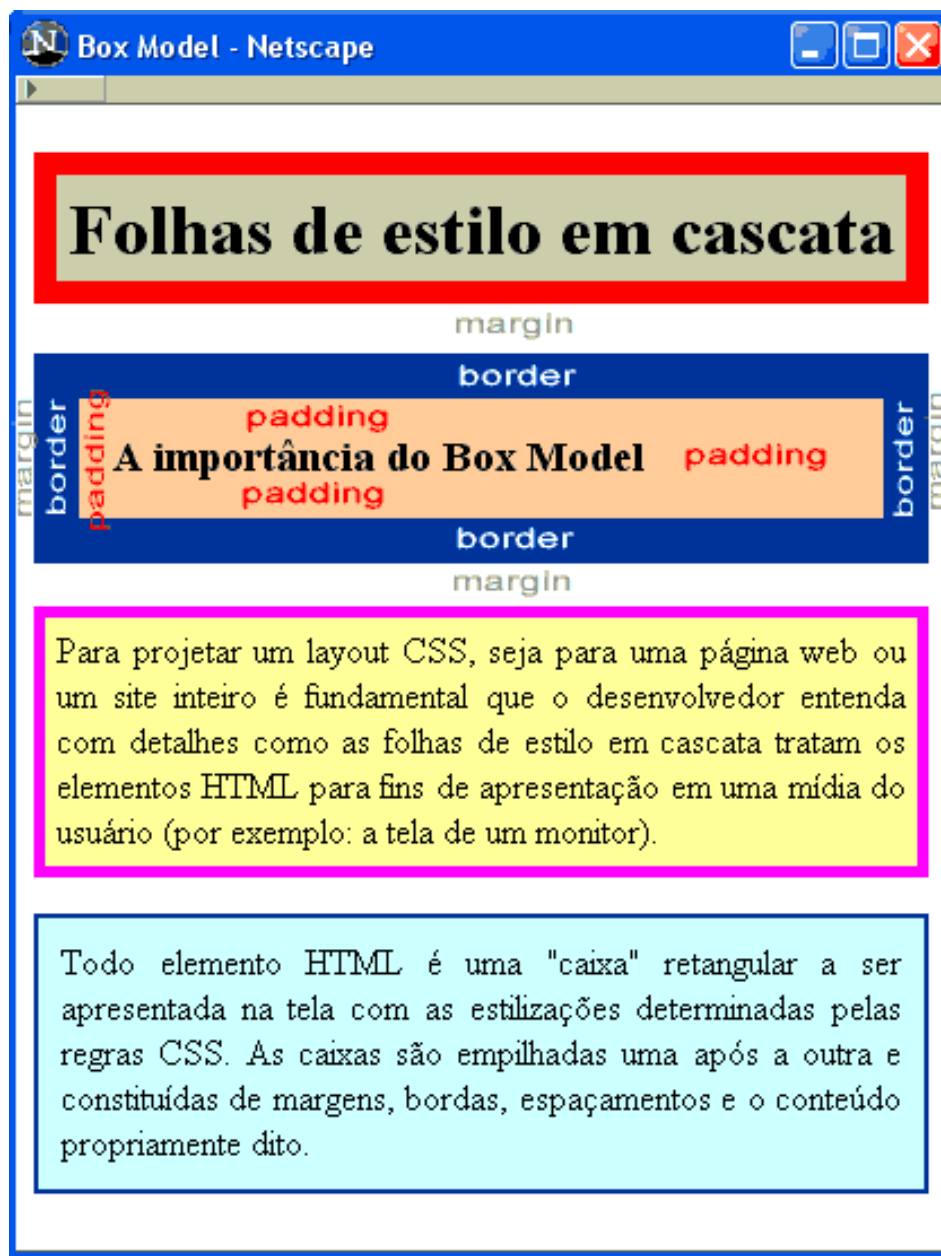
Links

- É possível criar formatações específicas para cada estado dos links:
 - `a:link` : estado inicial (não visitado)
 - `a:visited` : visitado
 - `a:hover` : com o mouse sobre o link
 - `a:active` : link ativo ("clicado").
- Ex:
`a:link` {text-decoration: none; color: blue;}
`a:visited` {text-decoration: none; color: blue;}
`a:hover` {text-decoration: underline;}
`a:active` {text-decoration: none; color: green;}

Prática 29

- Crie uma página com links para suas atividades.
- Os links devem ter as configurações:
 - a:link: sem decoração e cor azul {text-decoration: none; color: blue}
 - a:visited: sem decoração e cor azul
 - a:hover : cor de fundo azul e cor do texto branca
 - a:active : cor de fundo verde e cor do texto branca

Blocos

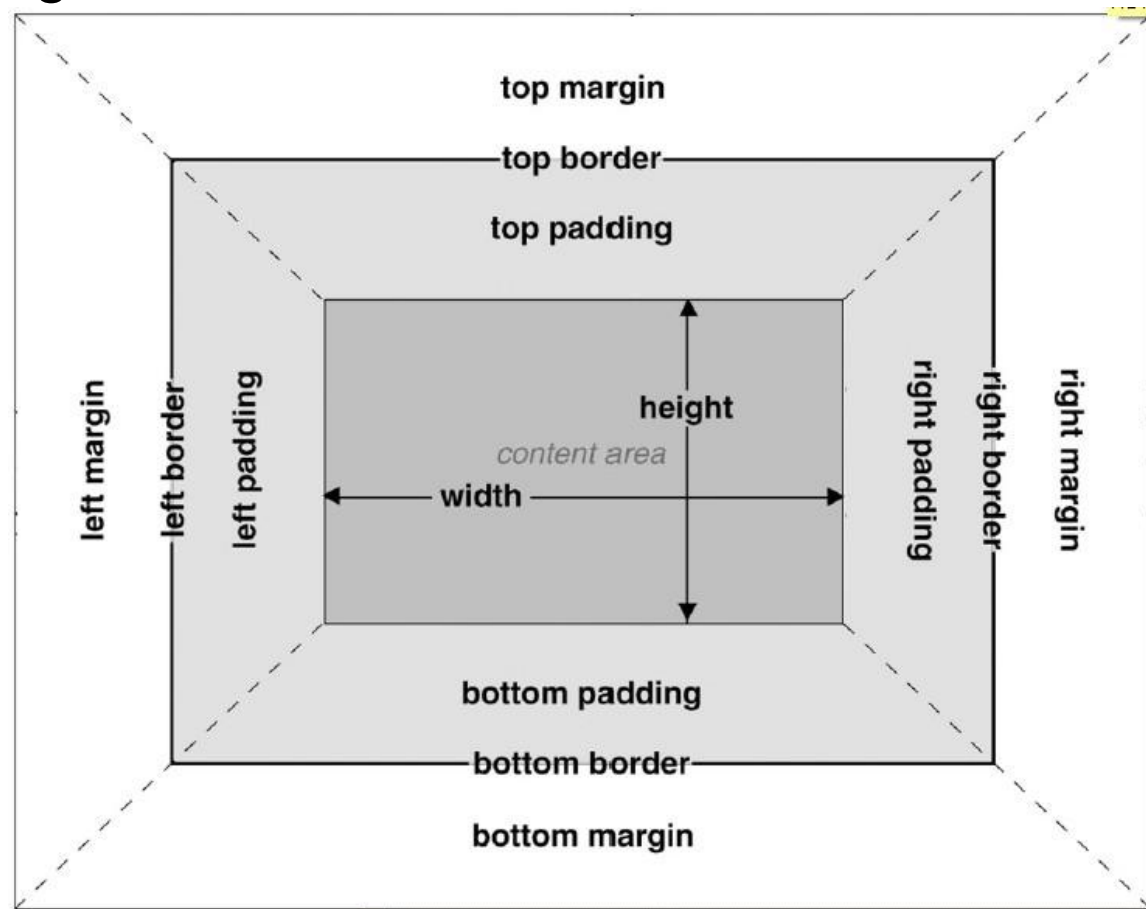


Tag div

- Têm importância fundamental em CSS
- Foi criada com o HTML 4 com a finalidade de fornecer um mecanismo genérico para agrupar e prover estrutura aos documentos
- **<div>...</div>** : tag que define um bloco ou seção da página e pode ser usada para dividir a o layout da página

Blocos

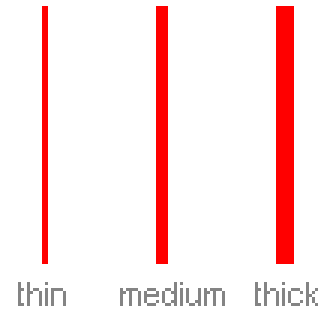
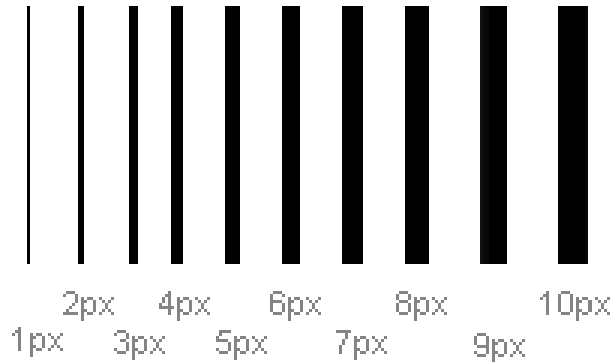
- Bordas, margens internas e externas possuem quatro lados: top, left, right e bottom
- Uma tag de bloco também possui uma largura e altura: width e height



Bordas

- **border-width**: espessura da borda.

– Ex: **border-width**: 10px ;

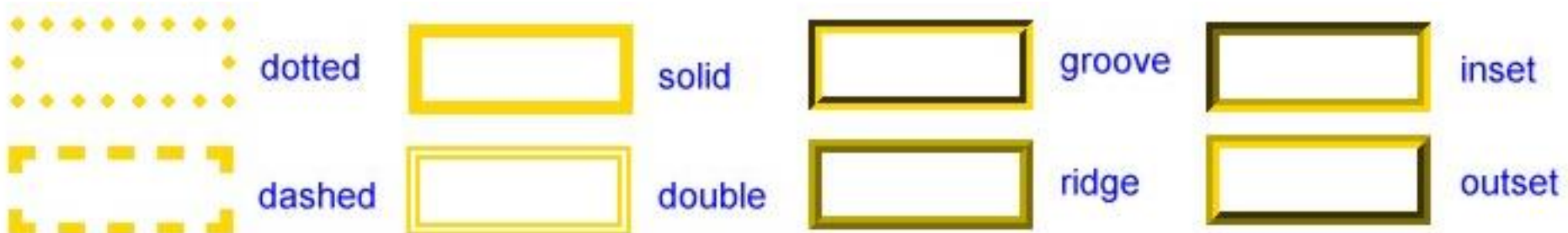


- **border-color**: cor da borda.

– Ex: **border-color**: #996699 ;

- **border-style**: estilo da borda.

– Ex: **border-style**: solid;



Bordas

- Valores individuais
 - **border-top-width**
 - **border-left-style**
 - **border-right-color**
 - **border-bottom-style**
 - **etc...**
- Ex: **border-top-color**: blue;
border-bottom-color: red;
- Forma reduzida:
 - **border**: *espessura estilo cor*
 - **border-top**: *espessura estilo cor*
 - **border-left**: *espessura estilo cor*
 - **etc...**
- Ex: **border**: 10px dotted #0000ff;
border-right: 5px double black;

Margens externas

- **margin**: define as margens **externas**. Os valores podem ser numéricos ou em %
- **margin-top**, **margin-bottom**, **margin-right** e **margin-left**: definem individualmente as margens **externas**.
- Ex:
margin: 10px;
margin: 20%;
margin-left: 10%;
margin-right: 10%;

ou ainda na forma reduzida:

margin: 10px 10px 10px 10px;

Em sentido horário: top,
right, bottom e left

Margens internas

- **padding**: define as margens **internas**. Os valores podem ser numéricos ou em %
- **padding-top**, **padding-bottom**, **padding-right** e **padding-left**: definem individualmente as margens **internas**.
- Ex:
padding: 10px;
padding: 20%;
padding-left: 10%;
padding-right: 10%

ou ainda na forma reduzida:

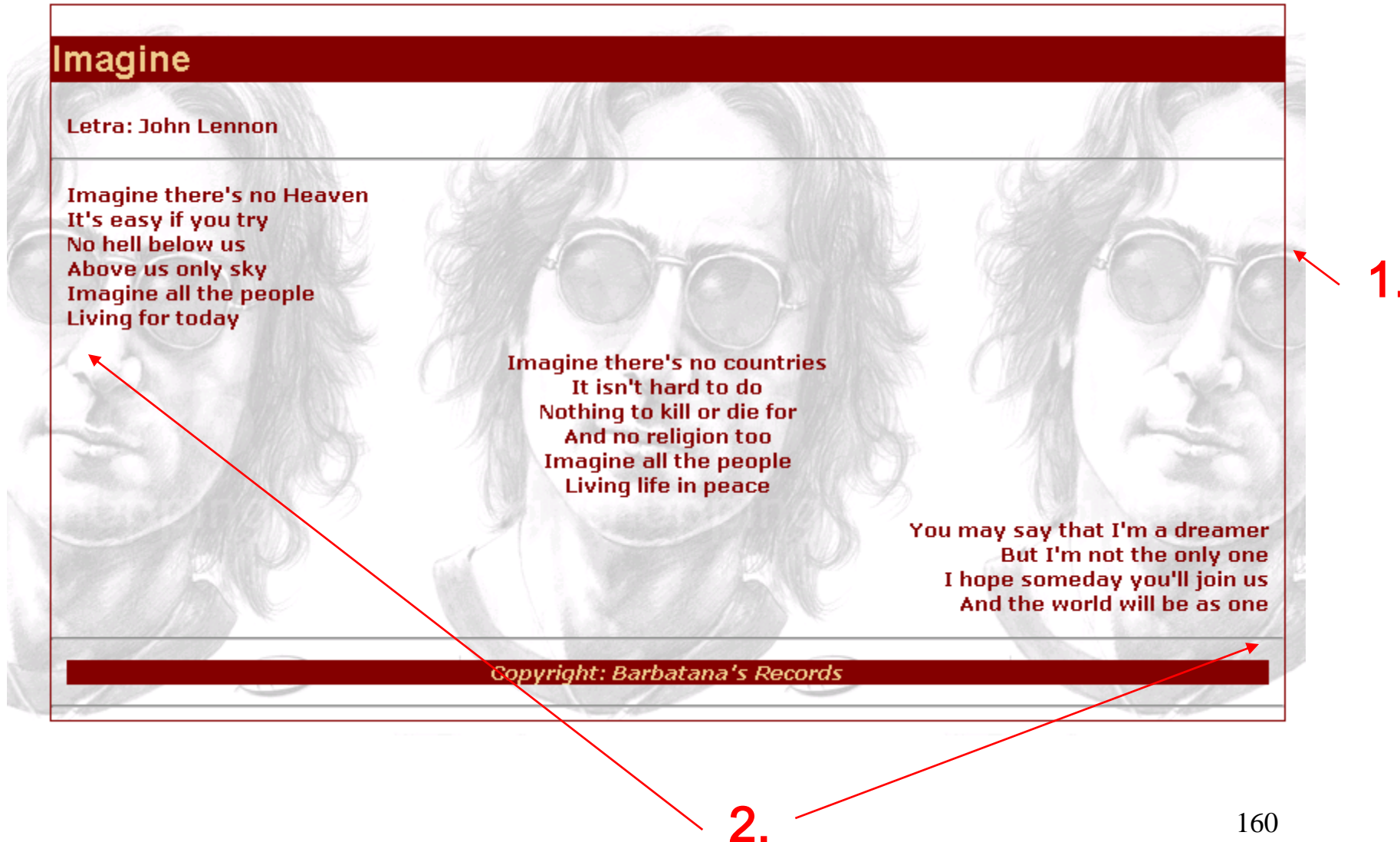
padding: 0% 10% 0% 10%



Em sentido horário: top, right, bottom e left

Prática 30

- Altere a página imagine.html de acordo com as instruções:



The image shows a screenshot of a web page titled "Imagine" with a background of John Lennon's face. The page has a dark red header with the title "Imagine" in yellow. Below the header, the text "Letra: John Lennon" is displayed. The main content area contains three stanzas of the song's lyrics. A red arrow labeled "1." points to the right side of the page. Another red arrow labeled "2." points to the bottom of the page, specifically to a dark red footer bar containing the text "Copyright: Barbatana's Records".

Imagine

Letra: John Lennon

Imagine there's no Heaven
It's easy if you try
No hell below us
Above us only sky
Imagine all the people
Living for today

Imagine there's no countries
It isn't hard to do
Nothing to kill or die for
And no religion too
Imagine all the people
Living life in peace

You may say that I'm a dreamer
But I'm not the only one
I hope someday you'll join us
And the world will be as one

Copyright: Barbatana's Records

1.

2.

Prática 30

1. Div envolvendo o conteúdo de body:
 - borda sólida, espessura de um pixel, cor marrom
 - margens externa da esquerda e da direita com 15%
2. Parágrafo com margens externa da esquerda e da direita com 15 pixels

Prática 31

- Altere a página sobre o filme Final Fantasy de acordo com as instruções:

Final Fantasy: The Spirits Within



1.

3. → Final Fantasy: The Spirits Within conta a história de uma Terra infestada de alienígenas no ano de 2065. Os humanos vivem em "cidades barreira", todos numa tentativa de livrar o planeta dos Phantoms (que significa Fantasma), uma misteriosa raça alienígena. A única esperança vem da cientista Aki Ross e de seu mentor, Dr. Sid, que tem um plano de destruir os Phantoms sem causar danos ao planeta, mas um general chamado Hein está determinado a usar o canhão espacial "Zeus" para destruir os Phantoms - mesmo que isso cause danos à Terra no processo.

2.

Copyright 2007 SQUARE ENIX CO., LTD. All Rights Reserved.

Prática 31

1. Div envolvendo o conteúdo de body:
 - borda pontilhada, espessura de um pixel, cor #ccb999
 - margens externa da esquerda e da direita com 15%
2. Parágrafo com margens externa da esquerda e da direita com 10 pixels
3. Span com texto em itálico

Layout em CSS

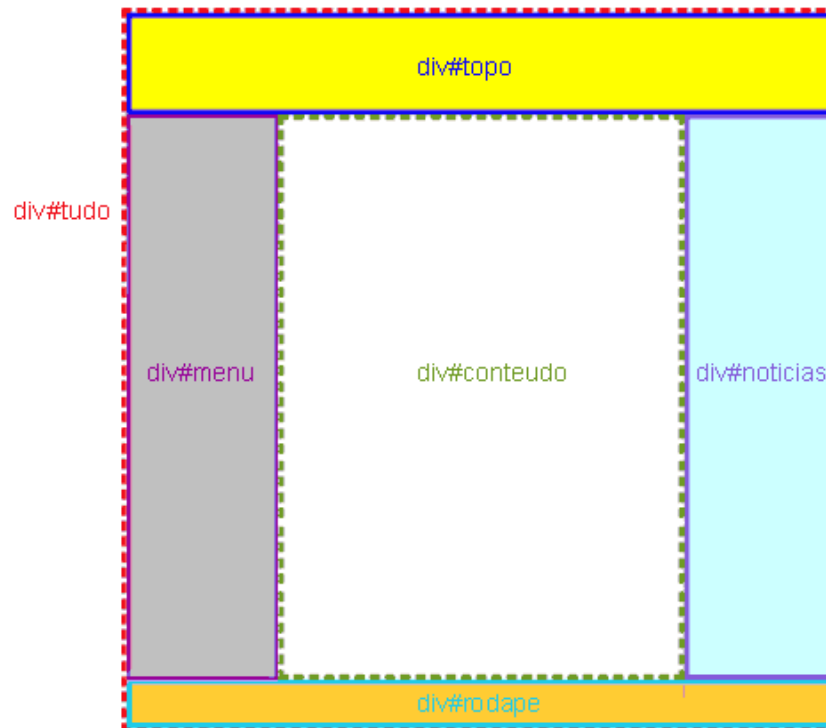
- Tipicamente se usam para layout tags `<div></div>` com CSS
- Tables continuam como opções válidas para dados tabulares
- Div x table:
 - Renderização:
 - Divs possuem renderização mais rápida
 - Tables somente são exibidas após a análise de toda a tabela, o que pode tornar lenta a exibição
 - Precisão: com divs a precisão ainda não é tão forte e temos que recorrer a truques para conseguir um layout e deixá-lo compatível com os navegadores

Layout com Divs

- Alguns desafios a superar:
 - Alinhamentos
 - Precisão com medidas relativas %
 - Divs que não acompanham o “esticamento” da página
- Esses desafios muitas vezes são superados com “truques” ou códigos hack.
- Ferramentas como DreamWeaver/Fireworks ajudam, mas ainda não chegaram ao nível de interação e fidelidade de visualização das tabelas

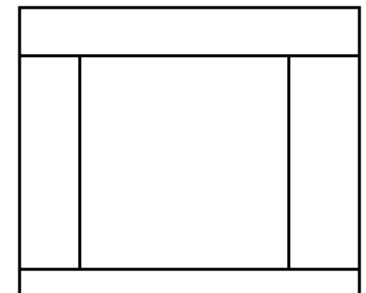
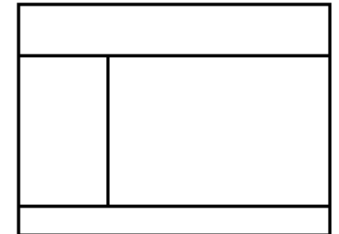
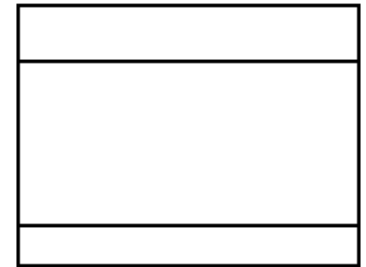
Layouts com Divs

- Tipicamente, um layout com divs inclui:
 - Uma tag div geral envolvendo todo o conteúdo
 - Outras tags div “fatiando” as outras seções
 - Tamanho fixo de colunas e demais divisões



Alguns layouts

- Alguns exemplos clássicos:
 - 1 coluna: cabeçalho, conteúdo e rodapé
 - 2 colunas: cabeçalho, menu esquerda, conteúdo e rodapé
 - 3 colunas: cabeçalho, menu esquerda, conteúdo, menu direita e rodapé



Layout simples com 1 coluna

```
...  
<div id="geral">  
  <div id="cabecalho">  
    ...  
  </div>  
  <div id="conteudo">  
    ...  
  </div>  
  <div id="rodape">  
    ...  
  </div>  
</div>  
...
```

```
#geral {  
  width: 700px;  
  margin: auto;  
  background-color: %ffffff;  
  color: #333333;  
  border: 1px solid gray;  
}  
  
#cabecalho, #conteudo, #rodape {  
  padding: 10px;  
}  
  
#cabecalho {  
  background-color: #dddddd;  
  border-bottom: 1px solid gray;  
}  
  
#conteudo {  
  border-top: 1px solid gray;  
}  
  
#rodape {  
  color: #333333;  
  background-color: #dddddd;  
  border-top: 1px solid gray;  
}
```


Layout simples com 1 coluna

Cabeçalho...

Um bom local para se descrever/apresentar a página e colocar uma imagem de principal.

Conteúdo

O conteúdo é o local onde você coloca os textos, imagens da sua página...É o desenrolar das suas idéias

Título 1

In nibh odio, porttitor vel, mattis vel, porttitor eget, orci. Duis consectetur tempus enim. Sed ut felis et tortor feugiat malesuada. Nam vulputate, tellus eu nonummy dapibus, pede risus luctus dolor, ut nonummy urna felis a lacus. Nam in dolor tincidunt ante vehicula dignissim. Nulla id libero ut dui placerat semper. Nulla sit amet nibh at velit facilisis gravida. Phasellus in diam. Pellentesque volutpat scelerisque dolor. Donec venenatis nulla suscipit nisi dignissim tristique. Ut ut lorem sit amet purus ullamcorper mattis. Phasellus ornare ligula non orci. Pellentesque luctus tristique orci. Proin a est. Nulla porta lacus ut leo. Nulla faucibus pede sit amet dui. Nullam aliquet lorem at nunc. Nullam eu tellus.

Título 2

Sed tempus turpis vel quam molestie pulvinar. Suspendisse venenatis dolor semper ipsum. Quisque tempus erat ac mi. Aliquam semper, est nec hendrerit dignissim, ligula turpis sagittis purus, ut viverra velit eros at augue. Pellentesque mi nisi, porta eget, pharetra ac, sollicitudin sit amet, nisi. In sapien ligula lorem sollicitudin facilisis, sodales eget, tempus in, mauris. Cras risus sem, adipiscing non, convallis ac, consectetur eu, dolor. In quam lorem ipsum dolor sit amet. Curabitur tempus aliquam nulla. Etiam eros.

Rodapé: Pode-se colocar links e informações de contato e outras coisas que você achar necessário

© Todos os direitos reservados | [Web Designer](#)

Layout com 2 colunas

```
...
<div id="geral">
  <div id="cabecalho">
    ...
  </div>

  <div id="esquerda">
    ...
  </div>

  <div id="conteudo">
    ...
  </div>
  <div id="rodape">
    ...
  </div>
</div>
...
```

```
#geral {
  width: 700px;
  margin: auto;
  background-color: #ffffff;
  color: #333333;
  border: 1px solid gray;
}

#cabecalho, #esquerda, #conteudo, #rodape {
  padding: 10px;
}

#cabecalho {
  background-color: #dddddd;
  border-bottom: 1px solid gray;
}

#esquerda {
  float: left; width: 150px;
}

#conteudo {
  float: left; width: 500px;
  border-left: 1px solid gray;
}

#rodape {
  clear: both; color: #333333;
  background-color: #dddddd;
  border-top: 1px solid gray;
}
)
```

Layout com 2 colunas

Cabeçalho...

Um bom local para se descrever/apresentar a página e colocar uma imagem de principal.

Nessa lateral, pode-se colocar um menu ou algo que sirva de navegação

- [Link 1](#)
- [Link 2](#)
- [Link 3](#)
- [Link 4](#)
- [Link 5](#)
- [Link 6](#)
- [Link 7](#)

Conteúdo

O conteúdo é o local onde você coloca os textos, imagens da sua página...É o desenrolar das suas idéias

Título 1

In nibh odio, porttitor vel, mattis vel, porttitor eget, orci. Duis consectetur tempus enim. Sed ut felis et tortor feugiat malesuada. Nam vulputate, tellus eu nonummy dapibus, pede risus luctus dolor, ut nonummy urna felis a lacus. Nam in dolor tincidunt ante vehicula dignissim. Nulla id libero ut dui placerat semper. Nulla sit amet nibh at velit facilisis gravida. Phasellus in diam. Pellentesque volutpat scelerisque dolor. Donec venenatis nulla suscipit nisl dignissim tristique. Ut ut lorem sit amet purus ullamcorper mattis. Phasellus ornare ligula non orci. Pellentesque luctus tristique orci. Proin a est. Nulla porta lacus ut leo. Nulla faucibus pede sit amet dui. Nullam aliquet lorem at nunc. Nullam eu tellus.

Título 2

Sed tempus turpis vel quam molestie pulvinar. Suspendisse venenatis dolor semper ipsum. Quisque tempus erat ac mi. Aliquam semper, est nec hendrerit dignissim, ligula turpis sagittis purus, ut viverra velit eros at augue. Pellentesque mi nisi, porta eget, pharetra ac, sollicitudin sit amet, nisi. In sapien ligula lorem sollicitudin facilisis, sodales eget, tempus in, mauris. Cras risus sem, adipiscing non, convallis ac, consectetur eu, dolor. In quam lorem ipsum dolor sit amet. Curabitur tempus aliquam nulla. Etiam eros.

Rodapé: Pode-se colocar links e informações de contato e outras coisas que você achar necessário

© Todos os direitos reservados | [Web Designer](#)

Layout 3 colunas

```
<div id="geral">
  <div id="cabecalho">
    ...
  </div>
  <div id="esquerda">
    ...
  </div>
  <div id="conteudo">
    ...
  </div>
  <div id="direita">
    ...
  </div>
  <div id="rodape">
    ...
  </div>
</div>
```

```
#geral{
  width: 800px; margin: auto;
  color: #333333;
  border: 1px solid gray;
}
#cabecalho, #esquerda, #conteudo, #rodape {
  padding: 10px;
}
#cabecalho {
  background-color: #dddddd;
  border-bottom: 1px solid gray;
}
#esquerda{
  float: left; width: 130px;
  background-color: #dddddd;
}
#conteudo {
  float: left; width: 470px;
  background-color: white;
}
#direita {
  float: right; width: 150px;
  background-color: #dddddd;
}
#rodape {
  clear: both; background-color: #dddddd;
  border-top: 1px solid gray;
}
```

Layout 3 colunas

Cabeçalho...

Um bom local para se descrever/apresentar a página e colocar uma imagem principal.

Nessa lateral, pode-se colocar um menu ou algo que sirva de navegação

- [Link 1](#)
- [Link 2](#)
- [Link 3](#)
- [Link 4](#)
- [Link 5](#)
- [Link 6](#)
- [Link 7](#)

Conteúdo

O conteúdo é o local onde você coloca os textos, imagens da sua página...É o desenrolar das suas idéias

Título 1

In nibh odio, porttitor vel, mattis vel, porttitor eget, orci. Duis consectetur tempus enim. Sed ut felis et tortor feugiat malesuada. Nam vulputate, tellus eu nonummy dapibus, pede risus luctus dolor, ut nonummy urna felis a lacus. Nam in dolor tincidunt ante vehicula dignissim. Nulla id libero ut dui placerat semper. Nulla sit amet nibh at velit facilisis gravida. Phasellus in diam. Pellentesque volutpat scelerisque dolor. Donec venenatis nulla suscipit nisl dignissim tristique. Ut ut lorem sit amet purus ullamcorper mattis. Phasellus ornare ligula non orci. Pellentesque luctus tristique orci. Proin a est. Nulla porta lacus ut leo. Nulla faucibus pede sit amet dui. Nullam aliquet lorem at nunc. Nullam eu tellus.

Título 2

Sed tempus turpis vel quam molestie pulvinar. Suspendisse venenatis dolor semper ipsum. Quisque tempus erat ac mi. Aliquam semper, est nec hendrerit dignissim, ligula turpis sagittis purus, ut viverra velit eros at augue. Pellentesque mi nisi, porta eget, pharetra ac, sollicitudin sit amet, nisi. In sapien ligula lorem sollicitudin facilisis, sodales eget, tempus in, mauris. Cras risus sem, adipiscing non, convallis ac, consectetur eu, dolor. In quam lorem ipsum dolor sit amet. Curabitur tempus aliquam nulla. Etiam eros.

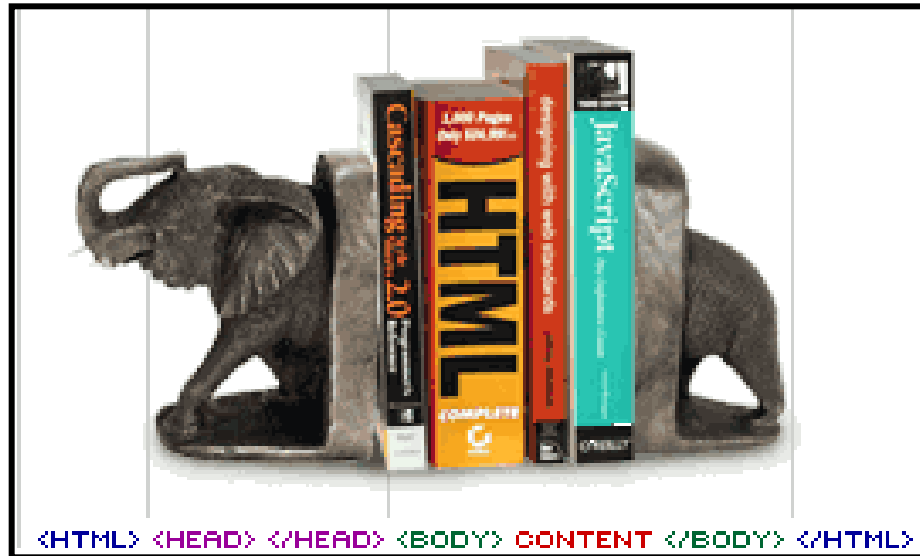
- [Sed accumsan congue](#)
- [Nulla dignissim nec augue](#)
- [Nunc ante elit nulla](#)
- [Aliquam suscipit consequat](#)
- [Cursus sed arcu sed](#)
- [Nulla dignissim nec augue](#)
- [Sed accumsan congue](#)
- [Nulla dignissim nec augue](#)
- [Nunc ante elit nulla](#)
- [Nunc ante elit nulla](#)
- [Aliquam suscipit consequat](#)
- [Cursus sed arcu sed](#)
- [Nulla dignissim nec augue](#)
- [Sed accumsan congue](#)
- [Nulla dignissim nec augue](#)
- [Nunc ante elit nulla](#)
- [Aliquam suscipit consequat](#)
- [Cursus sed arcu sed](#)

Rodapé: Pode-se colocar links e informações de contato e outras coisas que você achar necessário

© Todos os direitos reservados | [Web Designer](#)

Ainda sobre layouts

- Caso haja propriedades padding, border e margin deve-se considerar eventuais desajustes nas divs e corrigir as propriedades width
- Pode-se também utilizar a propriedade height fixas para padronizar o tamanho das divs
- Há ainda uma técnica chamada Faux Collums que consiste em colocar uma imagem “por trás” das divs que devem ser prolongadas
- Sobre os nomes esquerda, direita...:
 - Devem ser usados nomes que representem mais fielmente os trechos do layout, pois um dia pode-se pensar em mudar a ordem das divisórias e os nomes esquerda e direita estariam inválidos semanticamente



Programação Web Front-End

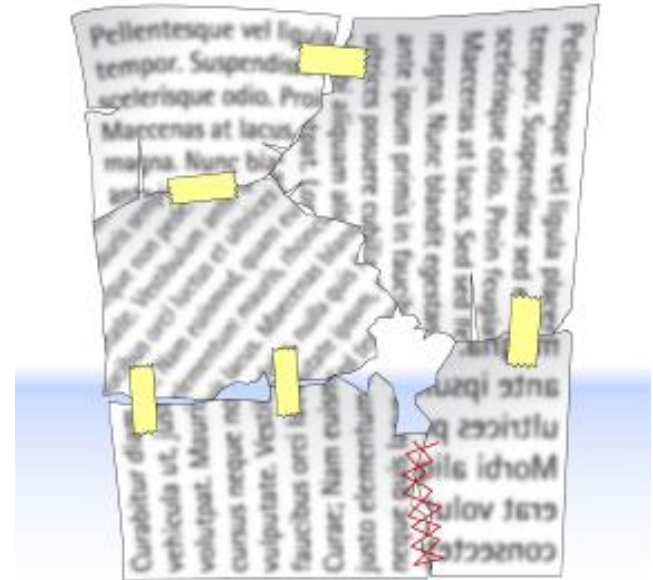
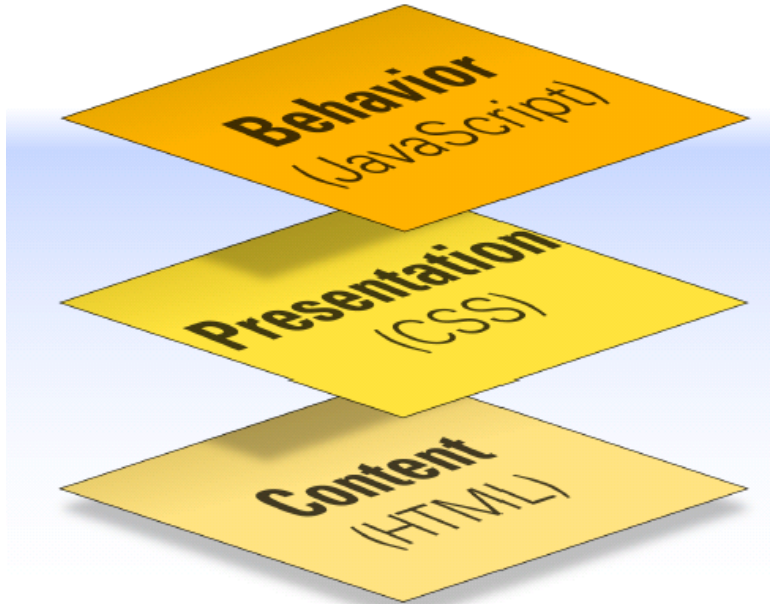
Introdução ao JavaScript

Introdução

- O código em uma página pode ser concebido em três visões distintas:
 - Estrutura e conteúdo: **HTML**
 - Apresentação: **CSS**
 - Comportamento: **JavaScript**
- Vantagens:
 - Reuso de partes do projeto
 - Modularidade
 - Flexibilidade e facilidade de manutenção
 - Legibilidade
 -

Introdução

- As 3 visões de concepção de uma página nos dão a visão em camadas, em vez de um código “macarronado” ou “remendado”



Características

- É uma linguagem poderosa, com sua utilização do lado cliente (browser)
- É uma linguagem de scripts que permite interatividade nas páginas web
- É incluída na página HTML e interpretada pelo navegador
- É simples, porém pode-se criar construções complexas e criativas
- JavaScript não é Java. São linguagens com aplicações e recursos totalmente distintos

Algumas coisas que se pode fazer com JS

- Validar entrada de dados em formulários: campos não preenchidos ou preenchidos incorretamente poderão ser verificados
- Realizar operações matemáticas e computação
- Abrir janelas do navegador, trocar informações entre janelas, manipular com propriedades como histórico, barra de status, plug-ins, applets e outros objetos
- Interagir com o conteúdo do documento tratando toda a página como uma estrutura de objetos
- Interagir com o usuário através do tratamento de eventos

Bibliotecas e frameworks

- **Prototype:** <http://www.prototypejs.org/>
- **script.aculo.us:** <http://script.aculo.us/>
- **Yahoo! User Interface Library (YUI):**
<http://developer.yahoo.com/yui/>
- **Dojo:** <http://dojotoolkit.org/>
- **jQuery:** <http://jquery.com/>
- **MooTools:** <http://mootools.net/>
- **Node.js :** <https://nodejs.org>
- **Angular.js :** <https://angularjs.org>
- **ReactJS :** <https://pt-br.reactjs.org/>

Formas de uso

- Dentro próprio código HTML:

```
<a href="#" onclick="alert('alo mundo!')">Diga alô</a>
```

- Separado em uma tag de script (preferencialmente dentro da tag `<head></head>`):

```
<script type="text/javascript">
```

```
    alert("alo mundo");
```

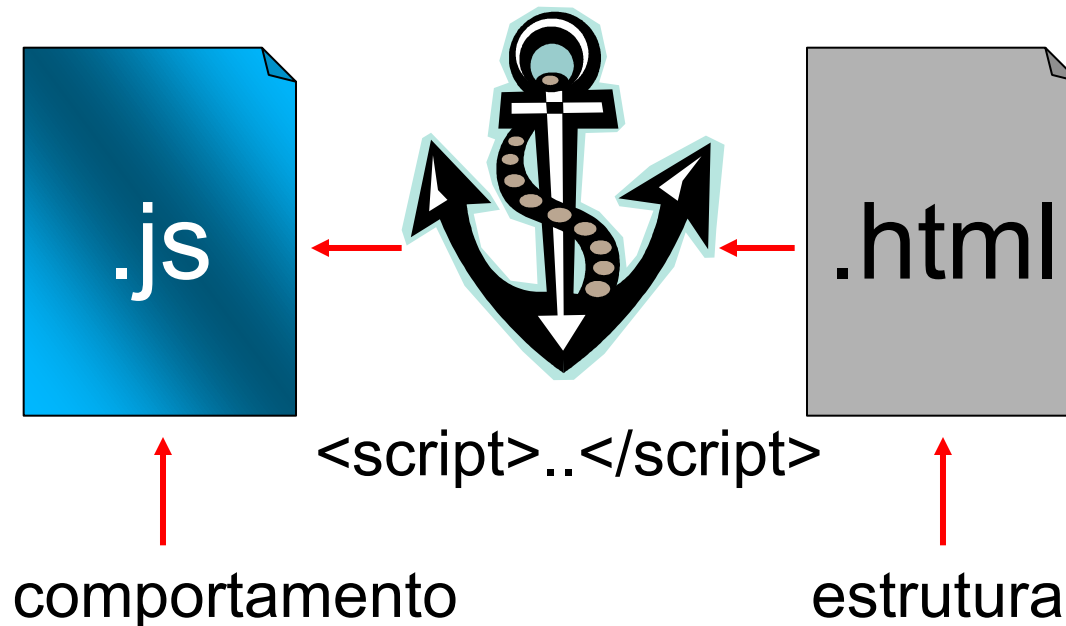
```
</script>
```

- Mais separado ainda dentro de um arquivo “texto” com extensão .js sendo chamado por uma tag script:

```
<script type="text/javascript" src="script.js"></script>
```

Dois arquivos separados?

- Usaremos dois arquivos texto:
 - Um com HTML com extensão .html
 - Outro com JavaScript com extensão .js
 - Haverá ainda uma tag HTML que “unirá” os arquivos



Alô mundo - versão 1

```
<html>
```

```
<head>
```

```
  <title>Alo mundo!</title>
```

```
  <script type="text/javascript" >
```

```
    alert("Alo mundo!");
```

```
  </script>
```

```
</head>
```

```
<body>
```

```
  ...
```

```
</body>
```

```
</html>
```

Alô mundo – versão 2

```
...  
<head>  
  <title>Alo!</title>  
  <script type="text/javascript" src = "alomundo.js">  
    </script>  
</head>  
...
```

alomundo.html

```
alert("alo mundo");
```

alomundo.js

Sintaxe

- Tudo é case-sensitive, ou seja: **teste** é diferente de **Teste**
- Construções simples: após cada instrução, finaliza-se utilizando um ponto-e-vírgula:
Instrução1;
Instrução2;
- Ex:
alert("alo");
alert("mundo");

Sintaxe

- Comentários de uma linha:

```
alert("teste"); // comentário de uma linha
```

- Comentário de várias linhas:

```
/* este é um comentário  
de mais de uma linhas */
```

- Saída de dados: em lugar de usar a função alert, podemos utilizar:

```
document.write("<h1>teste</h1>");
```

- Onde **document** representa a própria página e write escreve no seu corpo.

Variáveis

- Variáveis são usadas para armazenar valores temporários
- Usamos a palavra reservada **var** para defini-las
- Em JS, as variáveis são fracamente tipadas, ou seja, o tipo não é definido explicitamente e sim a partir de uma atribuição (=)
- Ex:

var x = 4; ← Declaração e atribuição de valor

var y; ← Declaração sem atribuição

y = 2; ← Atribuição

alert (x + y);

Alguns tipos

- Números: inteiros e decimais:

var i = 3;

var peso = 65.5;

var inteiroNegativo = -3;

var realNegativo = -498.90;

var expressao = 2 + (4*2 + 20/4) - 3;

- Strings ou cadeia de caracteres:

var nome = "josé";

var endereco = "rua" + " das flores"; ← concatenação

nome = nome + " maria"; ← concatenação

endereco = "rua a, numero " + 3; ← concatenação com
conversão numérica
implícita

Alguns tipos

- Lógico: tipo que pode ter os valores **true** ou **false**

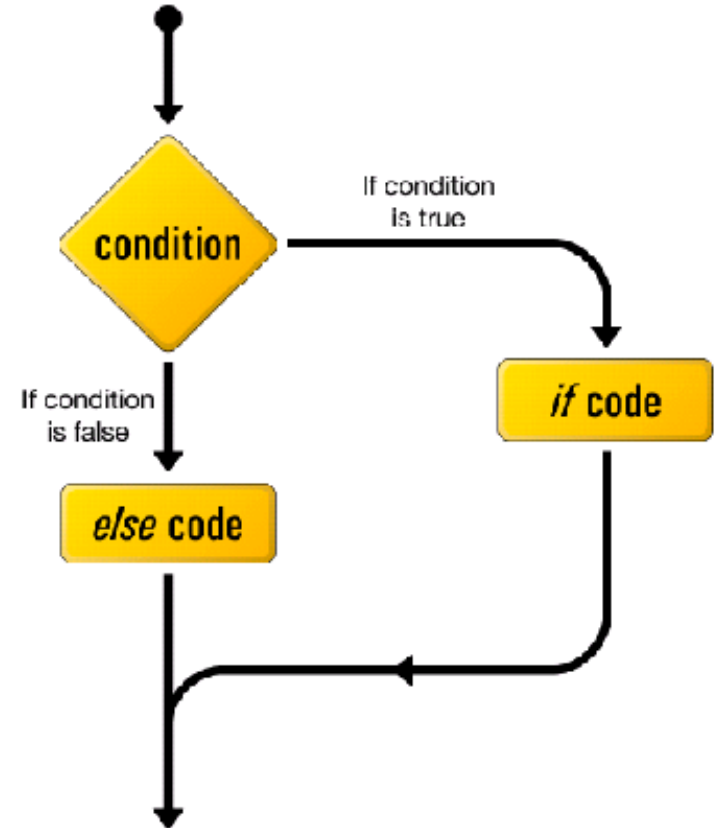
```
var aprovado = true;
```

```
alert(aprovado);
```

Estruturas de decisão – if e else

- Sintaxe:

```
if (condição) {  
    código da condição verdadeira;  
}  
else {  
    código da condição falsa;  
}
```



{ simboliza um início/begin
}

{ representa um fim/end

Operadores condicionais e lógicos

>	A > B
>=	A >= B
<	A < B
<=	A <= B
==	A == B
!=	A != B

← A é **igual** a B

← A é **diferente** de B

- && : and
- || : or
- ! : not

```
var idade = 17;  
if (idade >= 16 && idade < 18) {  
    alert("voto facultativo");  
}  
else {  
    alert("voto obrigatório");  
}
```

Estruturas de decisão – if e else

```
if (navigator.cookieEnabled) {  
    alert("Seu navegador suporta cookies");  
} else {  
    alert("Seu navegador não suporta cookies");  
}
```


Estruturas de decisão – Switch

```
switch (expressão) {  
    case valor 1:  
        //código a ser executado se a expressão = valor 1;  
        break;  
    case valor 2:  
        //código a ser executado se a expressão = valor 2;  
        break;  
    ...  
    case valor n:  
        //código a ser executado se a expressão = valor n;  
        break;  
    default:  
        //executado caso a expressão não seja nenhum dos valores;  
}
```

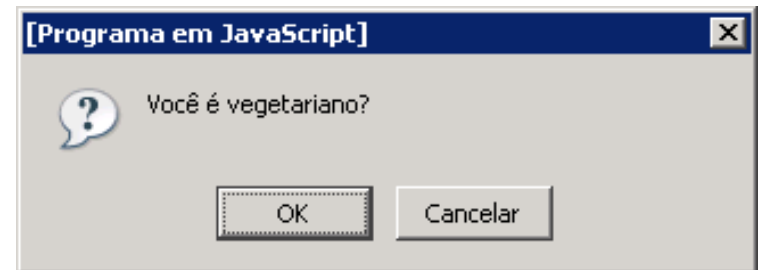
Estruturas de decisão – Switch

```
var idade = 20;  
switch (idade) {  
  case 29:  
    alert("Você está no auge.");  
    break;  
  case 40 :  
    alert("A vida começa aqui.");  
    break;  
  case 60 :  
    alert("Iniciando a melhor idade.");  
    break;  
  default:  
    alert("A vida merece ser vivida, não importa a idade.");  
    break;  
}
```

Janelas de diálogo - Confirmação

- Nos permite exibir uma janela pop up com dois botões: **ok** e **cancel**
- Funciona como uma função:
 - Se o usuário clicar em **ok**, ela retorna **true**; em **cancel** retorna **false**
- Ex:

```
var vegetariano = confirm("Você é vegetariano?");  
if (vegetariano == true) {  
    alert("Coma mais proteínas");  
}  
else {  
    alert("Coma menos gordura");  
}
```

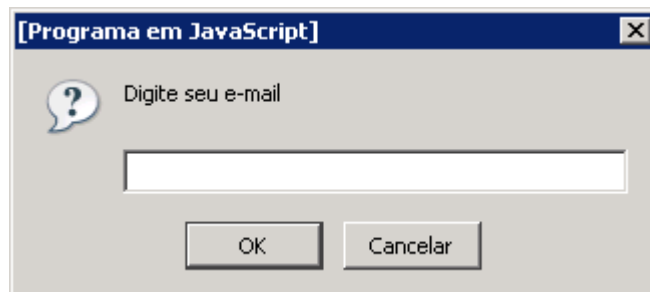


Janelas de diálogo - Prompt

- Nos permite exibir uma janela pop up com dois botões (**ok** e **cancel**) e uma caixa de texto
- Funciona como uma função: se o usuário clicar em **ok** e **prencher a caixa de texto**, ela retorna **o valor do texto**; em **cancel** retorna **null**
- O segundo parâmetro pode ser preenchido como uma sugestão
- Ex:

```
var email = prompt("Digite seu e-mail", "");
```

```
alert("O email " + email + " será usado para spam.");
```



Janelas de diálogo - Prompt

- O que lemos da janela prompt é uma string
- Podemos converter strings para inteiro utilizando as funções pré-definida **parseInt** e **parseFloat**
- **parseInt(valor, base)**: converte uma string para inteiro.
 - O valor será convertido para inteiro e base é o número da base (2,10,16...)
- **parseFloat(valor)**: converte uma string para um valor real/ponto flutuante

Janelas de diálogo - Prompt

- Ex:

```
var notaStr = prompt("Qual a sua nota?","");  
var trabalhoStr = prompt("Qual o valor do trabalho?","");  
var nota = parseFloat(notaStr);  
var trabalho = parseFloat(trabalhoStr);  
var media = (nota + trabalho) / 2;  
alert("Sua média é: " + media );
```

Estruturas de repetição - for

- Executa um trecho de código por uma quantidade específica de vezes
- Sintaxe:

```
for (inicio; condicao; incremento/decremento) {  
    //código a ser executado.  
}
```

- Ex:

```
var numeros = [1, 2, 3, 4, 5];  
for (var i = 0; i < numeros.length; i++) {  
    numeros[i] = numeros[i]* 2;  
    document.write(numeros[i] + "<br/>");  
}
```

Estruturas de repetição - while

- Executa um trecho de código enquanto uma condição for verdadeira
- Sintaxe:

```
while (condicao) {  
    //código a ser executado  
}
```

Ex:

```
var numero = 1;  
while (numero <= 5) {  
    alert("O número atual é: " + numero);  
    numero = numero + 1;  
}
```


Estruturas de repetição – do...while

- Executa um trecho de código enquanto uma condição for verdadeira
- Mesmo que a condição seja falsa, o código é executado pelo menos uma vez
- Sintaxe:

```
do {  
    //código a ser executado.  
} while (numero <= 5) ;
```

Ex:

```
var numero = 1;  
do {  
    alert("O número atual é: " + numero);  
    numero = numero + 1;  
} while (numero <= 5) ;
```

Expressões compactadas

- Em JS podemos utilizar formas “compactada” instruções:

numero = numero + 1 equivale a numero++

numero = numero - 1 equivale a numero--

numero = numero + 1 equivale a numero += 1

numero = numero - 1 equivale a numero -= 1

numero = numero * 2 equivale a numero *= 2

numero = numero / 2 equivale a numero /= 2

Práticas

- Elabore scripts usando a função prompt que:
 - 32. Leia um valor e imprima os resultados: “É maior que 10” ou “Não é maior que 10” ou ainda “É igual a 10”
 - 33. Some dois valores lidos e imprima o resultado
 - 34. Leia 2 valores e a operação a ser realizada (+, -, * ou /) e imprima o resultado (use um switch)
 - 35. Leia um nome e um valor **n** e imprima o nome **n** vezes usando o laço **for**

Funções

- Funções são blocos de código reutilizáveis.
- Elas não são executadas até que sejam **chamadas**
- Podem ter parâmetros de entrada e de saída
- Podemos ter vários parâmetros de entrada separados por vírgulas
- Podemos retornar um valor através da instrução **return**

Funções

- Sintaxe:

```
function nomeDaFuncao() {  
    //códigos referentes à função.
```

```
    ...
```

```
}
```

```
function nomeDaFuncao(p1, p2, p3, ...) {  
    //códigos referentes à função.
```

```
    ...
```

```
}
```

```
function nomeDaFuncao(p1, p2, p3, ...) {
```

```
    ...
```

```
    return p1+p2-p3;
```

```
    ...
```

```
}
```

Funções

- Ex. 1:

```
...  
<a href = "#" onclick = "alo();" >Chamar a função</a>  
...
```

← alomundo.html

```
function alo() {  
    alert("Link clicado!");  
}
```

← alomundo.js

Funções

- Ex. 2:

```
...  
<form>  
  <input type = "button" value = "Chamar função" onclick = "alo();" />  
</form>  
...
```

alomundo.html

```
function alo() {  
  alert("Link clicado!");  
}
```

alomundo.js

Funções

- Ex. 3: Passando parâmetros

```
...  
<form>  
  <input type = "button" value = "Chamar função"  
    onclick = "saudacao('jose');"/>  
</form>  
...
```

← saudacao.html

```
function saudacao(nome) {  
  alert("Olá, " + nome);  
}
```

← saudacao.js

Funções

- Ex. 4: Passando parâmetros de campos de formulário

```
...  
<form>  
  <input type="text" name="txtNome" id = "txtNome"/>  
  <input type="button" name="btn_saudacao"  
    onclick = "saudacao(document.getElementById('txtNome').value);"/>  
</form>  
...
```

```
...  
<form name = "frm">  
  <input type="text" name="txtNome"/>  
  <input type="button" name="btn_saudacao"  
    onclick = "saudacao(frm.txtNome.value);"/>  
</form>  
...
```

Funções

- Ex. : retornando valores e escrevendo no documento

```
function soma(v1, v2) {  
    return v1 + v2;  
}
```

← soma.js

```
function soma(v1, v2) {  
    document.write(v1 + v2);  
}
```

Eventos

- São reações a ações do usuário ou da própria página

ou:

- São ocorrências ou acontecimentos dentro de uma página. Ex:
 - Carregar uma página;
 - Clicar em um botão;
 - Modificar o texto de uma caixa de texto;
 - Sair de um campo texto;
 - etc;

Eventos

- **onclick**: ocorre quando o usuário clica sobre algum elemento da página

```
...  
<a href = "#" onclick = "alo();" >Chamar a função</a>  
...
```

- **onload** e **onunload**: ocorrem respectivamente quando o objeto que as possuem são carregados (criados) e descarregados

```
...  
<body onload = "bemvindo();" onunload = "adeus();" >  
...
```

Eventos

```
function bemvindo() {  
    alert("Seja bem vindo.");  
}
```

```
function adeus() {  
    alert("Obrigado pela visita.");  
}
```

Eventos

- **onmouseover**: é acionado quando o mouse se localiza na área de um elemento
- **onmouseout**: ocorre quando o mouse sai da área de um elemento

```
...  
<a href = "#" onmouseover = "mouseSobre();"   
      onmouseout = "mouseFora();" >
```

Passe o mouse

```
</a>
```

```
<div id = "resultado"> </div>
```

```
...
```

Eventos

```
function mouseSobre() {  
    var divResultado = document.getElementById("resultado");  
    divResultado.innerHTML = divResultado.innerHTML +  
        "mouse sobre.<br/>";  
}
```

```
function mouseFora() {  
    var divResultado = document.getElementById("resultado");  
    divResultado.innerHTML = divResultado.innerHTML +  
        "mouse fora.<br/>";  
}
```

Eventos

- **onsubmit**: usado para chamar a validação de um formulário (ao enviar os dados)
- Para validar um formulário, chamamos uma função por nós definida:
 - Ao chamar a função, usamos a palavra reservada return
- A função, por sua vez, deve retornar true ou false, representando se os dados devem ou não serem enviados. Ex:

```
<form name="frmBusca"
      action="http://www.google.com/search"
      method="get" onsubmit = "return validaCampo()">
  Termo: <input type="text" name="q" id = "q" />
  <input type="submit" name="btnBuscar" value="Buscar"/>
</form>
```


Eventos

```
function validaCampo() {  
    var valor = document.getElementById("q").value;  
  
    if ((valor == null) || (valor == "")) {  
        alert("Preencha o campo de busca");  
        return false;  
    }  
  
    return true;  
}
```

Eventos

- **onfocus**: ocorre quando um controle recebe o foco através do mouse ou do teclado
- **onblur**: ocorre quando um controle perde o foco

...

```
<input type="text" name="txt1" id = "txt1"  
      onfocus = "trataEntrada('txt1')"  
      onblur = "trataSaida('txt1')"/>
```

```
<input type="text" name="txt2" id = "txt2"  
      onfocus = "trataEntrada('txt2')"  
      onblur = "trataSaida('txt2')"/>
```

...

Eventos

```
function trataEntrada(id) {  
    var div = document.getElementById("resultado");  
    div.innerHTML = div.innerHTML + id + " ganhou o  
    foco.<br/>";  
}
```

```
function trataSaida(id) {  
    var div = document.getElementById("resultado");  
    div.innerHTML = div.innerHTML + id + " perdeu o  
    foco.<br/>";  
}
```

Eventos

- **onkeydown** e **onkeypress**: são semelhantes e ocorrem quando uma tecla é pressionada pelo usuário em seu teclado.
- **onkeyup**: é executado quando a tecla é liberada, ou seja, ela foi pressionada e em seguida liberada.

...

```
<input type="text" name="txtOrigem" id = "txtOrigem"  
      onkeydown = "copiaTexto('txtOrigem','txtDestino')"/>  
  
<input type="text" name="txtDestino" id = "txtDestino" />
```

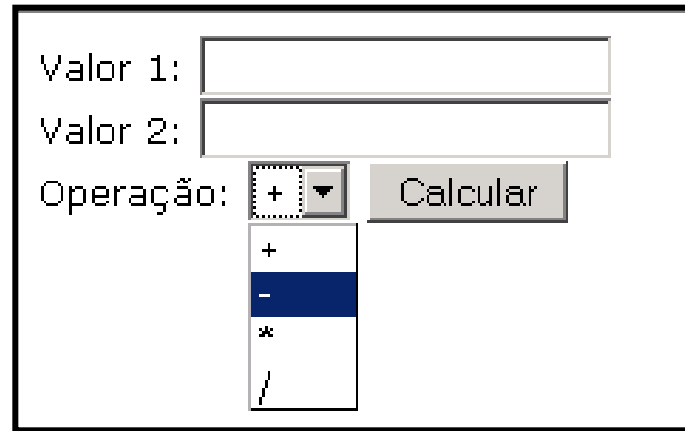
...

Eventos

```
function copiaTexto(idOrigem,idDestino) {  
    var txtOrigem = document.getElementById(idOrigem);  
    document.getElementById(idDestino).value =  
        txtOrigem.value;  
}
```

Prática 36

- Cria uma página semelhante à figura abaixo e implemente em JS uma calculadora com as 4 operações fundamentais



Valor 1:

Valor 2:

Operação:

+

-

*

/

▼

- O valor da caixa select poderá ser obtido da mesma forma que se obtém o valor das caixas de texto
- O resultado do cálculo deve ser exibido com uma função alert
- *Use a função `parseFloat` para converter números reais*

Prática 37

Elabore um formulário HTML que tenha como entrada 3 valores para lados de um triângulo e escreva uma função de nome **tipoTriangulo** que receba 3 parâmetros esses lados de um triângulo e imprima o tipo dele em uma div (equilátero, isósceles ou escaleno).

A passagem dos parâmetros deve ser feita de forma simplificada dentro do HTML no evento onclick de um botão ou link da seguinte forma:

```
<.... onclick = "tipoTriangulo(txtLado1.value, txtLado2.value, txtLado2.value)"...>
```

Prática 38

Deseja-se calcular a conta de consumo de energia elétrica de uma casa. Para isso, elabore um formulário em HTML que leia a quantidade de Kwh consumidos e o valor unitário do Kwh.

Escreva uma função em JavaScript que faça o cálculo (valor = quantidade x valor unitário) e, caso a quantidade de Kwh ultrapasse 100, o valor do Kwh deve ser acrescido em 25%. Caso ultrapasse 200, o mesmo valor deve ser acrescido em 50%.

Os valores devem ser repassados para uma função em JavaScript conforme a prática anterior.

Validações de formulários

- Os dados de um formulário devem ser enviados para um servidor.
- Pode-se suavizar o trabalho de um servidor efetuando-se algumas validações no próprio cliente (navegador) com JavaScript

– Nota:

É importante também haver a validação no servidor.

A validação com JavaScript serve apenas para amenizar o tráfego de rede com validações simples como campos não preenchidos, caixas não marcadas e etc.

Validações de formulários

- Algumas dicas:
 - Ao se validar um campo, procure sempre obtê-los pelo atributo `id`
 - Quase todos os elementos do formulário possuem sempre um atributo **`value`**, que pode ser acessado como uma `String`
 - As `Strings` possuem um atributo **`length`** que representa a quantidade de caracteres

Validações de formulários

- Validação de tamanho em campos de texto:
 - Pode-se limitar o campo a um tamanho mínimo ou máximo
 - Usa-se o atributo **length** para se checar o tamanho do campo valor do componente do formulário

Validações de formulários

- Validação de tamanho em campos de texto:

```
function validaCampoTextoTamanho(id, minimo, maximo) {  
    var valor = document.getElementById(id).value;  
    if (!validaCampoTexto(id)) {  
        return false;  
    }  
  
    if ( (valor.length < minimo) || (valor.length > maximo)) {  
        return false;  
    }  
    return true;  
}
```

Validações de formulários

- Validar se uma caixa de checagem (checkbox) está marcada:
 - Deve-se consultar o atributo **checked** do componente

```
function validaCampoCheckbox(id) {  
    var elemento = document.getElementById(id);  
    if (!elemento.checked) {  
        return false;  
    }  
    return true;  
}
```

Validações de formulários

- Validar se pelo menos um botão de radio de um conjunto foi selecionado:
 - Os campos radio funcionam em conjunto desde que possuam o mesmo atributo name, portanto não se deve consultar pelo id e sim pelo nome pelo método:

document.getElementsByName(nome)

- **getElementsByName(nome)** retorna um array de elementos com o mesmo nome.
 - Esse array deve ser percorrido verificando-se no atributo **checked** se pelo menos um dos botões de radio foram marcados
- Também use **document.getElementById(id)**

Validações de formulários

- Validar se pelo menos um botão de radio de um conjunto foi selecionado:

```
function validaCamposRadio(nome) {  
    var opcoes = document.getElementsByName(nome);  
    var selecionado = false;  
    for(var i = 0; i < opcoes.length; i++) {  
        if(opcoes[i].checked) {  
            selecionado = true;  
            break;  
        }  
    }  
    if(!selecionado) {  
        return false;  
    }  
    return true;  
}
```

Práticas

- Use uma página HTML e um arquivo de scripts
 - Use o evento **onsubmit** do formulário e uma função de validação que retorne **true** ou **false**
 - Utilize uma página qualquer como **action** do formulário.
39. Copie o valor de um campo texto para outro caso o campo de origem não esteja vazio. Use o evento onblur do campo de origem
40. Valide um campo senha de acordo com seu tamanho:
- a) < 3: segurança fraca
 - b) Entre 3 e 5: segurança média
 - c) >= 6: segurança forte
41. Valide se dois campos do tipo **password** são idênticos