

# Engenharia de Telecomunicações e Rede de Computadores



## **Desenvolvimento Web** Aula I

---

# **SOBRE A WEB**

---

# O que é Internet?



- A **Internet** é uma rede de computadores ao redor do mundo, onde cada computador pode se **comunicar** com outro.
- Na *Internet*, você pode **interagir** com pessoas de todo o mundo. Por outro lado, usando a *Intranet*, você pode interagir somente com pessoas de uma mesma organização.

# O que é a World Wide Web?

- ***A World Wide Web (WWW)***, conhecida como Web, é um sistema de informação que une dados de vários serviços de internet. Enquanto a internet refere-se aos componentes físicos da rede global, a Web refere-se ao corpo da informação compartilhada usando esta rede



# O que é linguagem de marcação?

- Páginas *Web* são os conteúdos que você vê nos sites. Estas páginas são armazenadas como arquivos com extensão **htm** ou **html**.
- Uma **linguagem de marcação** é a linguagem usada para formatar páginas *Web* com texto e informações separadamente. Isto assegura que as páginas *Web* serão acessíveis de qualquer parte do mundo usando qualquer tipo de *browser*.

# O que é HTML?

- ***Hypertext Markup Language (HTML)*** é uma linguagem de formatação usada para criar páginas *Web*.
- Um documento HTML contém:
  - O texto a ser **exibido**.
  - A informação sobre como o texto precisa ser **formatado**.
  - A informação sobre como o texto precisa ser **estruturado**.

# Hipertexto

- É um tipo de texto, **digital**, que permite a leitura **não linear**, através de links. Quando você clica num link, é redirecionado à página correspondente.

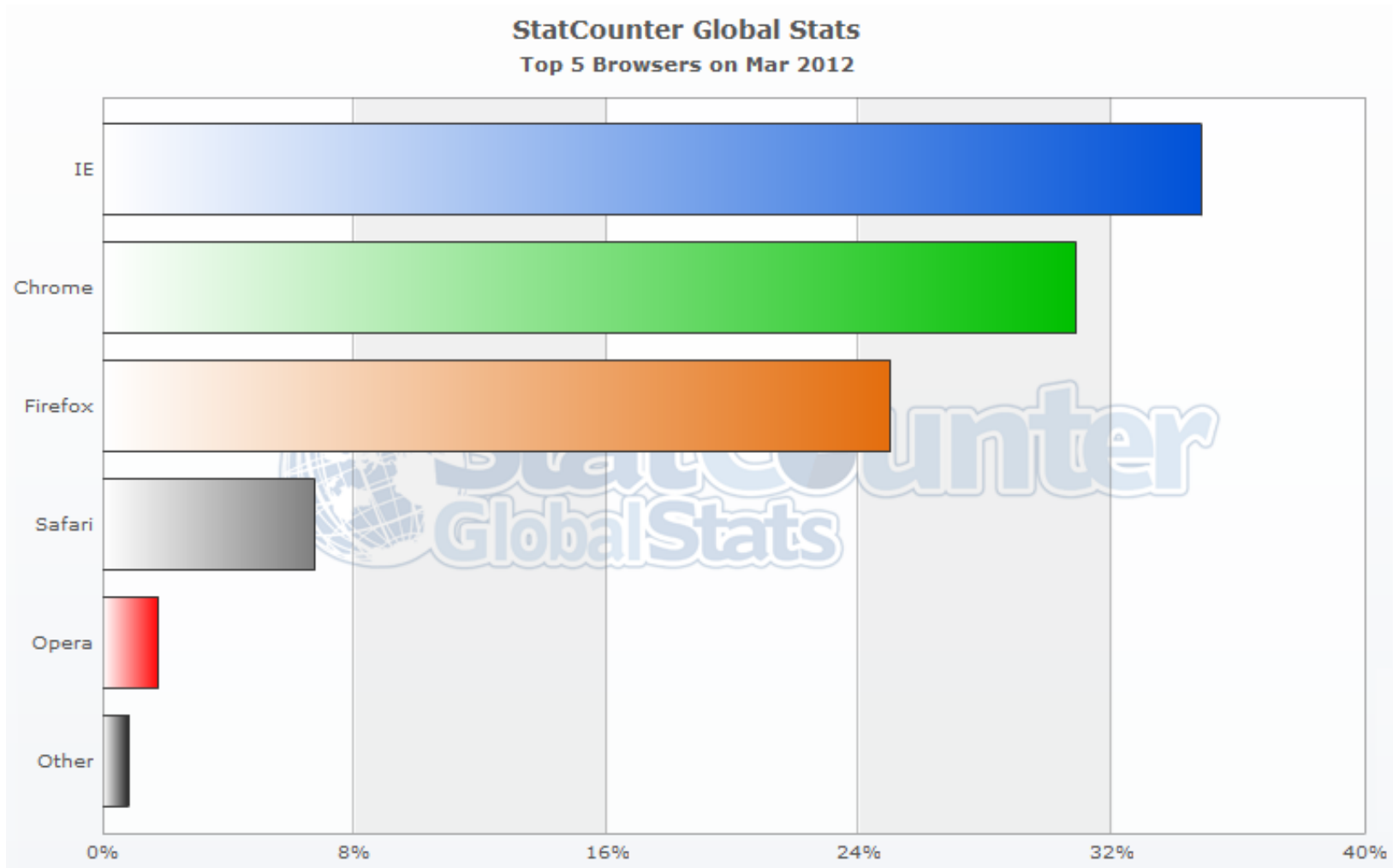
# Browsers

- São programas que você usa para **acessar**, **visualizar** e **interagir** com páginas *Web*. A *Web* permite que *browsers* de diferentes plataformas interpretem, formatem e exibam páginas *Web*.





# Ranking dos Browsers



# Servidores Web

- São **computadores robustos**, com programas que processam pedidos recebidos dos *browsers*. Estes servidores armazenam os web sites, que são uma coleção de páginas Web relacionadas entre si.
- Podem armazenar páginas estáticas ou dinâmicas.
  - **Páginas estáticas** terão sempre o mesmo comportamento.
  - **Páginas dinâmicas** geralmente estão ligadas a alguma [linguagem de programação](#).

# O que é uma home page?

- A **home page** é a primeira página de um site, ou a **página de entrada**. Por exemplo, quando você conecta ao site da Microsoft em [www.microsoft.com](http://www.microsoft.com), você pode não ser capaz de especificar um nome de arquivo. Nestes casos, o servidor Web exibirá um **arquivo padrão**. Este arquivo é a *home page*.
  - Uma homepage é normalmente chamada de **index.htm** ou de **default.htm** (os arquivos também podem possuir a extensão **.html** como padrão).

# HTTP, FTP e TCP

- Protocolos são um **conjunto de regras** para a comunicação em uma rede. Eles determinam como computadores em uma rede se comunicam, como começar, como manter e como terminar a comunicação.

# HTTP, FTP e TCP

- *Hyper Text Transfer Protocol* (HTTP) é o protocolo usado pelos servidores *Web* para transmitir documentos **HTML** pela Internet.

# HTTP, FTP e TCP

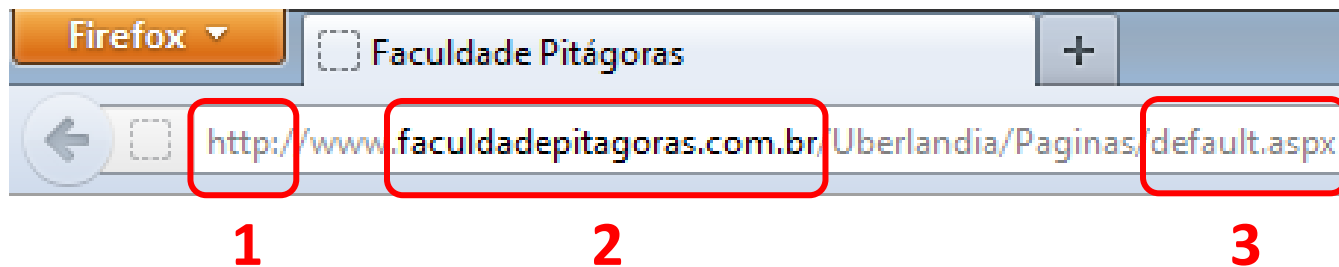
- *File Transfer Protocol (FTP)* é outro **protocolo** muito utilizado para transmitir arquivos entre computadores na *Web*, porém sua **interface gráfica** não é tão agradável como o HTTP.

# HTTP, FTP e TCP

- ***Transfer Control Protocol* (TCP)** é um **conjunto de protocolos** usado para a **transmissão de informação** pela Web, e possui a função de verificar se os dados são transferidos de forma correta, na sequência apropriada e sem erros.
- **IP** significa *Internet Protocol*, e é um protocolo de **endereçoamento**, ou seja, ele fornece o endereço dos computadores na rede.

# O que é uma URL?

- **Universal Resource Locator (URL)** é o sistema de endereçamento usado pelos *browsers* para localizar redes, computadores e arquivos na Internet.



1. Protocolo HTTP
2. Endereço do servidor
3. Nome do arquivo no servidor



---

# HTML E A WEB

---

# O que significa HTML?

- HTML significa *Hypertext Markup Language*. Uma linguagem de marcação é uma linguagem que você usa para **formatar informações** como uma entidade separada em um documento.

**<html></html>**

<head></head>

<p></p>

<title></title>

**<body></body>**

# O que significa HTML?

- Você usa HTML para criar páginas *Web*. HTML é baseado no *Standard Generalized Markup Language* (SGML).
- O SGML foi o primeiro **padrão internacional** para a troca de documentos usando qualquer **meio eletrônico**.
- As características principais do HTML são:
  - HTML é uma linguagem de formatação.
  - Documentos HTML contêm textos **ASCII** como seu elemento básico.

# Quais as vantagens do uso do HTML?

- Com o HTML você pode:
  - **Criar** facilmente documentos *Web*.
  - **Assegurar** a acessibilidade das suas páginas Web por um grande número de plataformas e *Web browsers*.
  - **Transmitir** facilmente documentos pela Internet.

Linkando

Criando

HTML

Acessando

Transmitindo

# Como criar um documento HTML?

- **HTML** utiliza um formato **simples** que pode ser gerado sem a ajuda de aplicações especiais.
- Para criar um documento HTML, tudo o que precisa é de um **editor de texto** [ASCII](#). Se você precisar verificar o formato do documento durante a criação, você pode usar qualquer *browser*.

# Como criar um documento HTML?

- O HTML foi a primeira linguagem de marcação que pôde atender um **grande número de plataformas de computadores e *browsers***.
- Além disso, com o HTML pode-se assegurar que seu documento será lido por um grande número de pessoas. Isto acontece por que todos os *browsers* são capazes de interpretar o código.

# Como criar um documento HTML?

- Este tipo de linguagem é adequada para a **transmissão de documentos** pela Internet para vários tipos de computadores, usuários e *browsers*. Você pode transmitir rapidamente páginas *Web* criadas usando o HTML.

---

# **CRIANDO UM DOCUMENTO HTML**

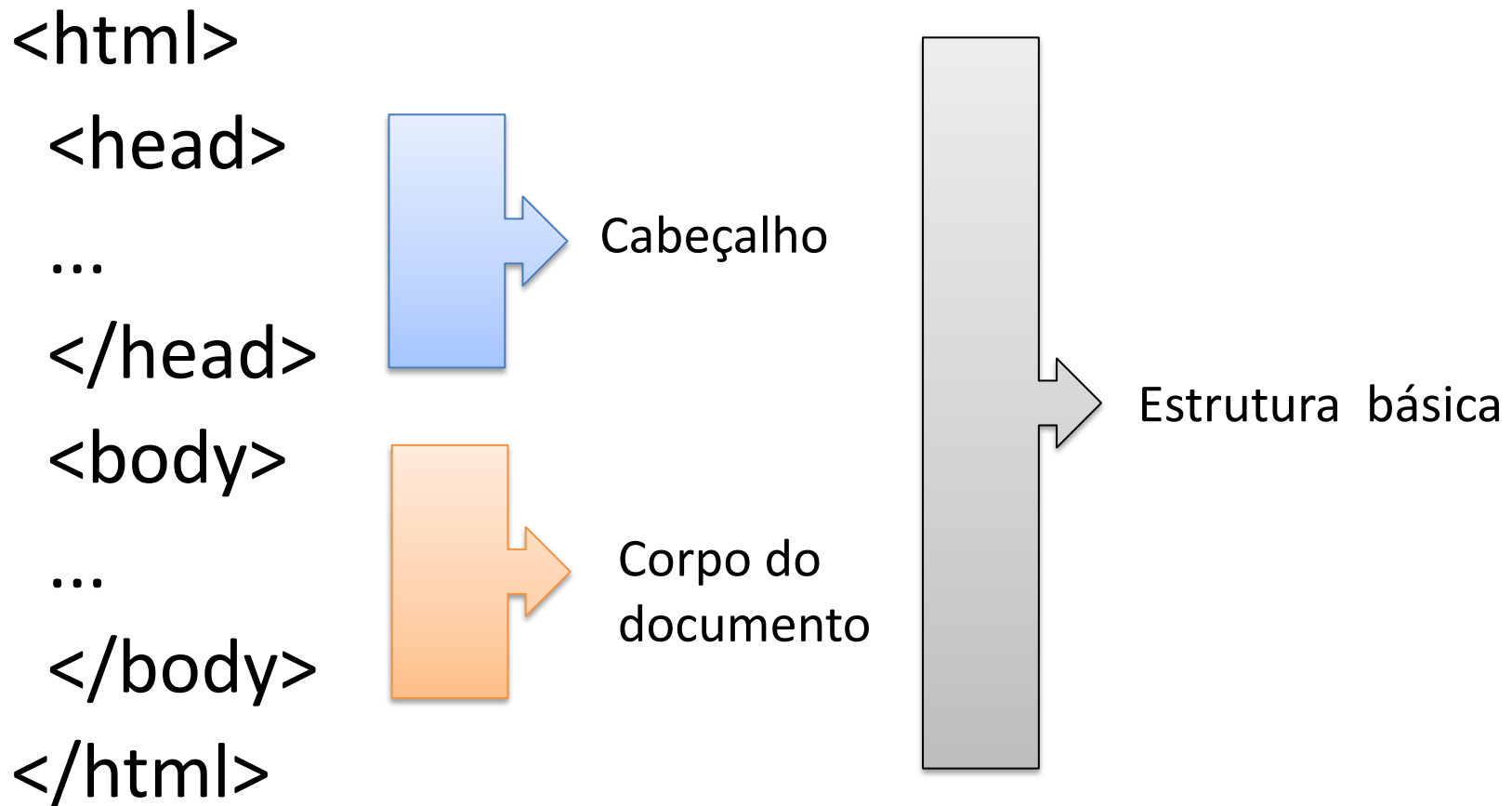
---



# Estrutura de um documento HTML

- Um documento HTML possui **duas partes**: *head* (cabeçalho) e *body* (corpo).
- No *head* você entra com o **título** da página *Web* e outras informações, que detalharemos mais adiante. No *body* você entra com todo o **conteúdo** da página *Web*.

# Estrutura de um documento HTML



# O papel das tags

- ***Tags*** são os elementos básicos de um documento HTML. Você usa *tags* para especificar o formato do conteúdo de uma página Web.
- As *tags* são denotadas por um único caractere ASCII ou uma *string* de caracteres entre os sinais < (menor que) e > (maior que), e geralmente são **palavras em inglês** ou simplesmente **abreviações** de palavras da língua norte-americana.

# O papel das tags

- Sinais de maior e menor permitem aos *browsers* **distinguir tags** de um texto comum. Por exemplo, um *web browser* irá interpretar, num documento HTML, como uma *tag* tudo que está entre < e > e como *body* (ou corpo da página) o texto que você quer exibir na página Web correspondente, necessariamente fora do espaço entre < e >.

# O papel das tags

- Grande parte das tags de HTML requerem seu **fechamento**: se você abre uma tag parágrafo `<p>`, você tem de fechá-la com `</p>`. Então, tudo que estiver compreendido entre `<p>` e `</p>` será encarado como o corpo do parágrafo. Todas as tags que necessitam de fechamento têm esta **mesma estrutura**: seu fechamento é idêntico à tag de abertura, porém com o sinal `/`.

# Tags com fechamento

- Por exemplo, você pode querer exibir um bloco de texto em negrito. No documento HTML, você usa a *tag* de início **<B>** no começo do bloco de texto e a de finalização **</B>** no fim. A *tag* ***Bold* (negrito)** é uma *tag* que requer fechamento.

# Tags sem fechamento

- Algumas outras tags **não necessitam de fechamento**, e sua sintaxe é somente de abertura. Por exemplo, a *tag* Break **<br>**, que insere uma quebra de linha, é uma delas.

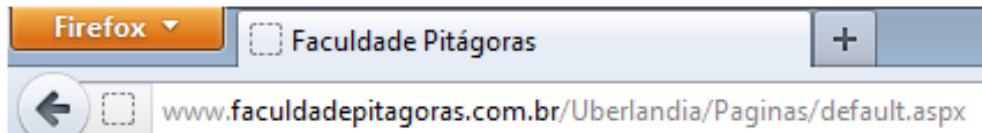
# Especificando o título da página

- Em qualquer documento HTML, você usa uma *tag* chamada **HTML** no começo do documento. Esta *tag* informa ao *browser* que o documento é um documento formatado em HTML.
- A parte do cabeçalho de um documento HTML contém o **título** da página *Web*. Para especificar o título, você usa a *tag* **TITLE** entre a de início e de finalização da *tag* **HEAD**.



# Especificando o título da página

- O browser **não exibe** nada que estiver dentro do **cabeçalho**, exceto o **título**. Outras informações como meta-tags e propriedades globais do documento não são visíveis para o usuário final.



```
<html>
<head>
<title> Faculdade Pitágoras </title>
</head>
<body>
</body>
</html>
```

# Usando a Tag BODY para criar o corpo de um documento HTML

- Depois da criação do cabeçalho de um documento HTML, você cria o **corpo** deste documento. O corpo contém o **texto**, os **objetos** que serão exibidos na página *Web* e o tipo da **formatação**.
- Você usa a *tag BODY* para criar o corpo de um documento HTML. A sintaxe desta *tag* está exibida na imagem ao lado. O *browser* exibe o texto, no formato especificado, que está incluído entre o início e o fim da *tag BODY*.

---

# **FORMATANDO UM DOCUMENTO HTML**

---

# Utilizando a tag parágrafo

- Você pode usar **parágrafos** para agrupar **linhas de texto** em uma página *Web*. O *browser* irá exibir o texto em um único parágrafo se você não criar outros.

```
1 |<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
  |"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
2 |<html xmlns="http://www.w3.org/1999/xhtml">  
3 |<head>  
4 |<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
5 |<title>Desenvolvimento Web</title>  
6 |</head>  
7 |<body>  
8 |<p>Isto é um parágrafo!</p>  
9 |</body>  
10|</html>
```

# Inserindo quebras de linhas em uma página Web

- Se você precisar exibir uma parte do texto em uma **nova linha**, você pode inserir uma quebra de linha utilizando a **tag *Break***, como mostra a imagem. O *browser* exibe a parte do texto escrito depois da **tag Break `<br/>`**, em uma nova linha.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Desenvolvimento Web</title>
6 </head>
7 <body>
8 <p>Isto é um <br />parágrafo!</p>
9 </body>
10 </html>
```

# Criando subtítulos

- Com o uso da *tag Heading* `<hX>` `</hX>` você pode incluir **títulos** e **subtítulos** em diferentes **blocos de texto** numa página, a fim de **aumentar a legibilidade**.
  - Por exemplo, se você precisar fornecer dois níveis de títulos em uma página, você pode usar o conjunto de tags `<h1>` `</h1>` e `<h2>` `</h2>`. Pode-se utilizar as *tags* `<h1>` `</h1>` para especificar o **título principal** e as *tags* `<h2>`,`</h2>` para o **subtítulo secundário**.

# Criando subtítulos

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
2 <html xmlns="http://www.w3.org/1999/xhtml">  
3 <head>  
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
5 <title>Desenvolvimento Web</title>  
6 </head>  
7 <body>  
8 <h1>Primeiro Nível do Título</h1>  
9 <h2>Segundo Nível do Título</h2>  
10 <h3>Terceiro Nível do Título</h3>  
11 </body>  
12 </html>
```

## Primeiro Nível do Título

## Segundo Nível do Título

## Terceiro Nível do Título

# Adicionando comentários

- Você pode adicionar **comentários** para incluir **notas, sugestões e explicações** que não fazem parte do documento. Você usa a *tag* ***Comment*** (comentário) `<!-- xxxxxx -->` para adicionar comentários a um documento HTML.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Desenvolvimento Web</title>
6 </head>
7 <body>
8 <!-- Isto é um comentário -->
9 </body>
10 </html>
```



# Usando Tag pré-formatada

- Podemos exibir **espaços múltiplos e linhas em branco** em uma página usando a *tag Pre-formatted* (Pré-formatado).

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Desenvolvimento Web</title>
6 </head>
7 <body>
8 <pre>Texto com espaços
9
10 múltiplos e linhas
11
12 em branco</pre>
13 </body>
14 </html>
```

Texto com espaços  
múltiplos e linhas  
em branco

# Exibindo um texto em negrito

- Você usa a *tag* **Bold (negrito)** `<b>` `</b>` para exibir um texto em negrito em uma página *Web*. A sintaxe é exibida na imagem.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Desenvolvimento Web</title>
6 </head>
7 <body>
8 <p><b>Texto em negrito!</b></p>
9 </body>
10 </html>
```

**Texto em negrito!**

# Exibindo um texto em itálico

- Outra opção é a **tag Italic (itálico)** `<i>` `</i>`, para exibir um texto em itálico numa página *Web*. Observe a imagem.

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5  <title>Desenvolvimento Web</title>
6  </head>
7  <body>
8  <p><i>Texto em itálico!</i></p>
9  </body>
10 </html>
```

*Texto em itálico!*

# Exibindo um texto sublinhado (underline)

- Usa-se a *tag* **Underline (Sublinhado)** `<u>` `</u>` para exibir um texto sublinhado em uma página *Web*.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Desenvolvimento Web</title>
6 </head>
7 <body>
8 <p><u>Texto em sublinhado!</u></p>
9 </body>
10 </html>
```

Texto em sublinhado!

---

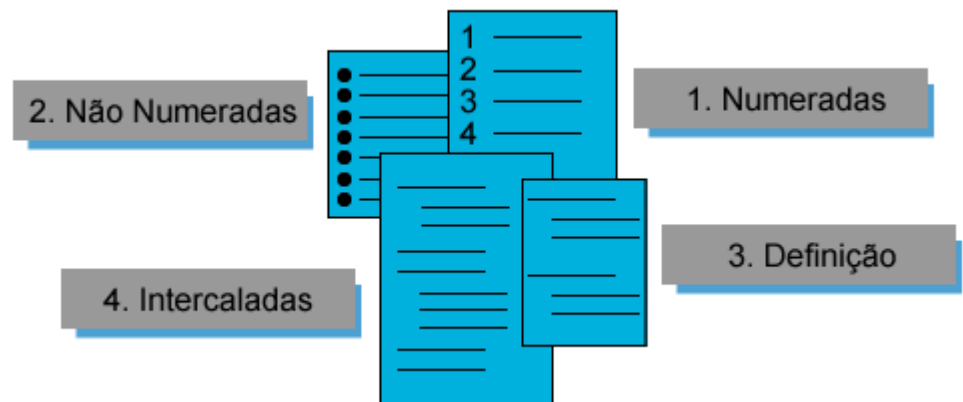
# **CRIANDO LISTAS**

---

# Tipos de listas

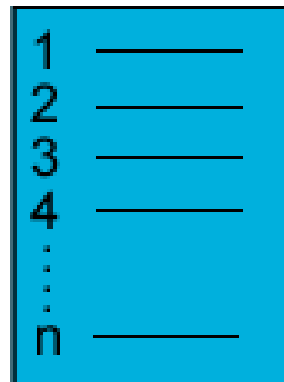
- Uma lista é uma coleção de **itens relacionados**. Você usa uma lista para **organizar dados** como uma sequência de passos ou itens em um grupo. O HTML fornece **quatro tipos**:

1. Numeradas
2. Não Numeradas
3. Definição
4. Intercaladas



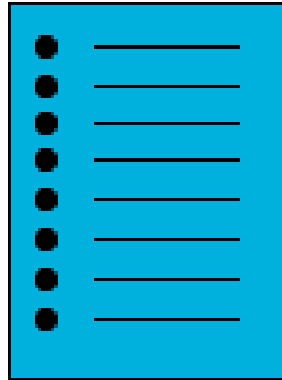
# Funções das listas

- Uma lista **numerada** é usada para exibir um conjunto de itens depois de um número ou uma letra.
  - Uma lista numerada é também chamada de **lista ordenada**.



# Funções das listas

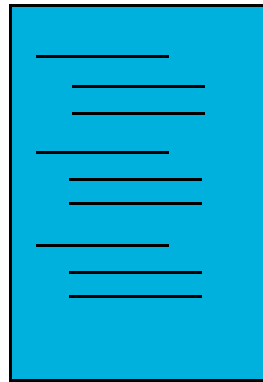
- Uma lista **não numerada** é usada para exibir um conjunto de itens depois de uma marcação.
  - Uma lista não numerada é chamada também de lista de marcação, ou ainda **não-ordenada**.





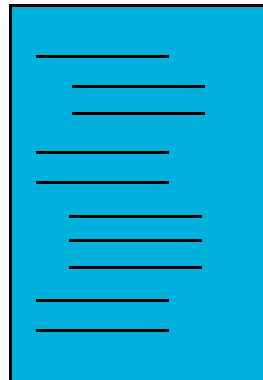
# Funções das listas

- Você usa uma **lista de definições** para exibir um conjunto de itens e suas descrições.



# Funções das listas

- Uma lista **intercalada** é uma **combinação de duas ou mais listas**. Você intercala a estrutura de uma lista com outra.
  - As listas intercaladas são muito pouco utilizadas hoje, por isso não entraremos em detalhes sobre seu uso e sua sintaxe.



# Criando listas ordenadas

- Uma lista ordenada é aquela que cada item é exibido depois de um número ou letra, numa sequência lógica. Sua sintaxe é exibida abaixo.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
2 <html xmlns="http://www.w3.org/1999/xhtml">  
3 <head>  
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
5 <title>Desenvolvimento Web</title>  
6 </head>  
7 <body>  
8 <ol>  
9 <li>Item 1</li>  
10 <li>Item 2</li>  
11 <li>Item 3</li>  
12 </ol>  
13 </body>  
14 </html>
```

1. Item 1  
2. Item 2  
3. Item 3

# Criando listas não numeradas

- A lista não numerada é utilizada quando **não há uma relação** ou estrutura específica entre os itens da lista. Sua sintaxe `<ul></ul>` é exibida na imagem acima. Este tipo de lista também pode ser chamada de **não-ordenada**.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Desenvolvimento Web</title>
6 </head>
7 <body>
8 <ul>
9 <li>Item 1</li>
10 <li>Item 2</li>
11 <li>Item 3</li>
12 </ul>
13 </body>
14 </html>
```

- Item 1
- Item 2
- Item 3

# O que significa lista intercalada?

- Também é possível criar **combinações** destas listas em uma página *Web*. A combinação de uma ou mais listas é chamada de **lista intercalada**, na qual estão alternadas a definição de uma lista com a definição da outra.

```
7 <body>
8 <li>Item Pai 1
9   <ul>
10     <li>Item 1
11     <li>Item 2
12   </ul>
13 <li>Item Pai 2
14   <ul>
15     <li>Item 1
16     <li>Item 2
17   </ul>
18 </body>
```

- Item Pai 1
  - Item 1
  - Item 2
- Item Pai 2
  - Item 1
  - Item 2

---

# ADICIONANDO GRÁFICOS

---

# Tipos de Imagens

- Neste módulo, veremos os **tipos de imagens** que você pode inserir em uma página *Web* e também **como inserir imagens** em uma página.
  - Você usa imagens para **melhorar a comunicação** dos conceitos explicados em uma página *Web*. Imagens melhoram a **aparência** da página e **destacam as ideias** apresentadas no documento.

# Adicionando Imagens

- Você utiliza imagens para incluir **títulos** e **logos** em página. Você pode usar também como *links* para outras páginas *Web*.
- Quando adicionamos uma imagem a uma página *Web*, é preciso manter em mente o **tamanho, formato e posição** desta imagem na página.

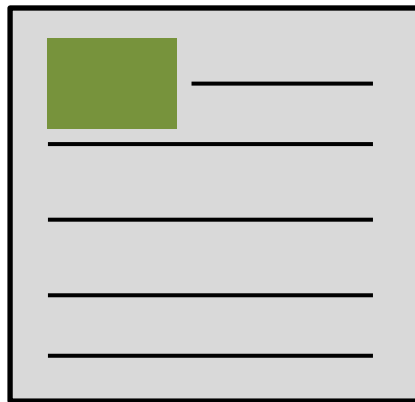


# Criando imagens

- Quando criamos uma imagem para uma página Web, é preciso observar o **tamanho do arquivo**, para que pessoas com todos os tipos de conexões possam ter acesso. Lembre-se: quanto menor a **velocidade de conexão** do usuário, mais tempo ele irá demorar para carregar a imagem.
- Você pode arrumar as imagens de várias maneiras e posições em sua página. Entretanto, o **posicionamento** de diversas imagens na mesma página deverá ser consistente.

# Alinhando uma imagem

- Você pode **alinhar** uma imagem em uma página *Web* com:
  - Outros elementos usados na página, como o texto ou outras imagens;
  - Bordas da **janela** do *browser*.



# Alinhando uma imagem

- O formato que você usar para uma imagem determina sua aparência. As primeiras imagens da Web eram formatadas utilizando entre **16 e 256 cores**, já que este era o limite máximo de exibição dos monitores. Além disso, o uso de um número grande de cores numa imagem elevava muito o tamanho do arquivo. Atualmente não existe uma preocupação muito grande com a **quantidade de cores**, mas sim com o **tamanho do arquivo**.

# Formatos de imagens

- *A World Wide Web* suporta quase todos os tipos de **arquivos de imagens**. Entretanto, somente um número limitado de formatos é utilizado. Veremos os três mais utilizados.
  - GIF
  - JPEG ou JPG
  - PNG

# Formatos de imagens

- **GIF** - *Graphics Interchange Format* foi introduzido pela CompuServe em 1987 e desde então foi **amplamente utilizado** na Web. Seu formato suporta oito bits por pixel, permitindo o uso de uma paleta de até **256 cores**. Utiliza o tipo de compressão LZW e é mais indicado para gráficos, ícones e imagens que não necessitam muitas cores. Permite imagens com **fundo transparente**.

# Formatos de imagens

- **JPEG ou JPG** - *Joint Photographic Experts Group* é o método **mais utilizado** para comprimir **imagens fotográficas**. Foi introduzido em 1983 e suporta até **16,8 milhões de cores**. Não permite imagens com fundos transparentes.

# Formatos de imagens

- **PNG** - *Portable Networks Graphics* é o **substituto do formato GIF**. No início de 1995 o padrão de compressão LZW foi patenteado pela empresa Unisys, que anunciou cobrança de royalties para o uso deste tipo de arquivo. Esta informação, aliada à crescente capacidade de exibição de cores dos computadores, levou à criação do PNG, que assim como JPG, permite exibição de até **16,8 milhões de cores** e, como o GIF, **fundos transparentes**.

# Utilizando a *tag IMAGE*

- A tag img é utilizada para inserir uma imagem em uma página Web.

```

```



Tag utilizada para inserir uma imagem na página web.



Atributo usado para especificar o arquivo da imagem a ser inserida.



Nome / endereço do arquivo que contém a imagem a ser inserida.



# Usando ALT, ALIGN e TITLE

- Além do atributo **SRC**, você pode especificar outros atributos com a *tag Image* (imagem). Os atributos que você pode usar são **ALT**, **ALIGN** e **TITLE**.

```

```



# Usando ALT E ALIGN

- Em caso do browser não conseguir exibir a imagem por alguma razão, foi inventado o atributo ALT, que nada mais é que um **texto alternativo** à imagem. Este atributo é também considerado de **acessibilidade**, já que é valioso para **deficientes visuais**: programas para estas pessoas leem as imagens e o ALT pode fornecer um contexto para a imagem.

# Especificando o alinhamento da imagem

- O atributo **ALIGN** é utilizado para especificar o **alinhamento** da imagem em relação ao texto na página *Web*. Você pode alinhar o texto no topo, no centro ou no rodapé da imagem.
- Para alinhar um texto no topo da imagem, use o valor **TOP**, para o centro, **MIDDLE** e para o rodapé, **BOTTOM**. Além disso, a imagem pode ser alinhada à esquerda – **LEFT** – e à direita – **RIGHT**.

---

# **CRIANDO LINKS**

---

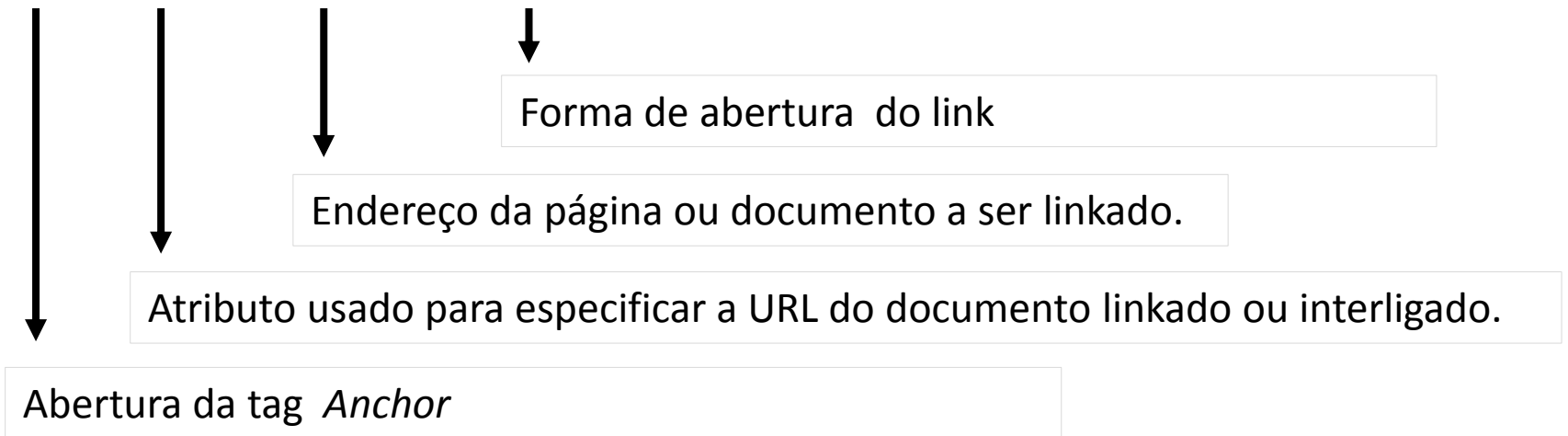
# Sobre os Links

- **Links** são conexões pelas quais você conecta páginas que podem ser do mesmo website, externas, de qualquer outro site, ou ainda conectar páginas a outros **documentos**.
- Os links podem ser criados no **texto** e também em outros **elementos** do website, como em **imagens** ou itens de um menu.

# Usando a *tag* **ANCHOR**

- A *tag* **Anchor** `<a>` `</a>` (âncora) é utilizada para criar *links* no hipertexto. A **sintaxe** desta *tag* é exibida na imagem abaixo.

```
<a href="URL" target="_blank">Texto descritivo</a>
```



# Criando *links* em textos

- Abaixo você pode verificar um **exemplo** de documento **HTML** com o **código** para a criação de um *link* no texto "Site de Buscas", direcionando para o Google, que é atualmente maior site deste tipo de serviço.

```
<a href="http://www.google.com.br/">Site de Buscas</a>
```

[Site de Buscas](http://www.google.com.br/)

# Criando *links* em textos

- O browser exibe o **hiperlink** em um formato diferente do texto normal, normalmente **sublinhado** e na cor **azul**, como na imagem da página anterior.
- Ao especificar uma URL apropriada para um serviço de Internet, você também pode linkar ou interligar sua página *Web* a esse serviço.
  - Por exemplo, você pode criar um ***link*** entre sua página e o seu e-mail.
- Selecionando o hiperlink, o *browser* ativa o **programa de e-mail** que irá permitir a você enviar uma mensagem eletrônica ao endereço especificado.



# Criando *links* para uma seção na mesma página

- Utilizamos a *tag* ***Anchor*** (âncora) para criar **links** de seção. Especificamos a URL do documento que desejamos interligar com o nome da seção e com o símbolo # antes do nome.

# Criando *links* para uma seção na mesma página

- Utiliza-se o atributo **NAME** para **definir as seções** que foram interligadas ao hipertexto. Observe a **sintaxe** para a definição da seção.
- O *browser* exibe o *link* e, quando selecionado, exibirá a **seção** que começa com o **nome especificado**.

# Criando *links* para uma seção na mesma página

- Caso você possua uma página com um texto muito extenso, é possível fazer links para dentro do próprio texto ou ainda diferentes seções da **mesma página**.

```
<a href="#secao1">Seção 1</a><br />
<a href="#secao2">Seção 2</a>
<br />
<a name="secao1">
<h1>Esta é a seção 1 do documento</h1>
</a>
<a name="secao2">
<h1>Esta é a seção 2 do documento</h1>
</a>
```

# Criando *links* em imagens

- Usamos a combinação da *tag* ***Anchor*** (âncora) e da *tag* ***Image*** (imagem) para criar um *link* numa imagem. Observe a sintaxe para este tipo de operação.

```
<a href="URL"></a>
```

# Criando *links* em imagens

- Observamos então que para inserir um *link* numa imagem usamos dois **conceitos** aprendidos anteriormente: **inserção de *links*** e **inserção de imagens**.

---

# USANDO COMBINAÇÕES DE TAGS

---

# Combinações de tags

- Podemos utilizar **combinações de *tags*** em páginas *Web* para melhorar sua **aparência e funcionalidade**.
- Você pode combinar a *tag Anchor* com:
  - *Tag* de formatação
  - Tag List
  - *Tag Image*
  - Uma combinação da *tag Image* e *List*

# Tag anchor com tags de formatação

- Primeiramente, vamos combinar a **tag Anchor** (âncora – links) com as **tags de formatação**.
  - Para lembrar: a **tag Anchor** (âncora) é usada para interligar páginas *Web*. Por padrão, os *browsers* exibem o hiperlink **sublinhado**. Caso deseje **modificar o formato do link**, você pode combinar a **tag Anchor** com as **tags de formatação**, como a **tag Italics** (itálico), **Bold** (negrito) e colocar os *links* como itens de uma **lista**, seja ela **ordenada** ou não.

```
<a href="URL"><i>Link com itálico</i></a>
```

*Link com itálico*



# Tag anchor com tags de listas

- Em uma **lista de links**, você define cada **item** como um *link* usando a *tag Anchor*. Aqui, uma lista **não ordenada** é usada em combinação com a *tag Anchor*.

```
<ul>
<li><a href="empresa.html">Empresa</a></li>
<li><a href="servicos.html">Serviços</a></li>
<li><a href="clientes.html">Clientes</a></li>
</ul>
```

- [Empresa](#)
- [Serviços](#)
- [Clientes](#)

# Thumbnails

- Podemos imaginar então a possibilidade de construir uma **lista de imagens**, que contém **links** para outras **páginas, documentos**, ou até mesmo outras **imagens**.
- Um recurso comum na internet é a utilização de ***thumbnails***, que são imagens em tamanhos reduzidos.

```
<ul>
<li><a href="servicos.html"></a></li>
<li><a href="produtos.html"></a></li>
<li><a href="contato.html"></a></li>
</ul>
```



# Thumbnails

- Embora o **HTML** permita as tags **HEIGHT** e **WIDTH** para especificar, por exemplo, a **altura e largura de uma imagem**, este recurso não é o mais indicado, já que, mesmo reduzindo as dimensões da imagem na tela, o **tamanho do arquivo** continua o mesmo. O procedimento mais adequado aqui é utilizar um programa de **edição de imagens** para produzir uma versão da imagem com **tamanho reduzido**.

---

# USANDO IMAGENS MAPEADAS

---

# O que são imagens mapeadas?

- Neste módulo, veremos como incluir uma **imagem mapeada** em uma página *Web*.
- Uma imagem mapeada é uma imagem dividida em **diferentes áreas** e cada área é interligada a um documento diferente. Uma imagem mapeada é diferente de um *link* numa imagem.

# O que são imagens mapeadas?

- Em um *link* colocado em uma imagem, o *browser* identifica a **URL** do documento especificado na *tag Anchor* e exibe a página interligada. No caso da imagem mapeada, o *browser*, primeiramente, identifica a **área da imagem** que você selecionou e então a URL do **documento vinculado** a esta área.

# O que são imagens mapeadas?

- O exemplo mais comum de uso é quando existe a necessidade de **uma mesma imagem** direcionar o usuário para **duas ou mais páginas web diferentes**, como num **menu**. Neste caso, delimitam-se áreas da imagem que remetem a um lugar e outras áreas para outras páginas ou documentos.

# Tipos de imagens mapeadas

- Você pode criar dois tipos de **imagens mapeadas** no HTML:
  - imagens mapeadas ***server-side*** (no servidor) e;
  - imagens mapeadas ***client-side*** (no cliente).



# Tipos de imagens mapeadas

- Na imagem mapeada server-side (no servidor), você define um arquivo especial chamado **arquivo de definição das coordenadas** com todos os detalhes da imagem mapeada e armazena este arquivo no servidor. Quando você seleciona uma **área** da imagem mapeada, o browser acessa o arquivo de definição das coordenadas no servidor para identificar a **URL** do documento interligado. Neste momento trataremos apenas de imagens mapeadas do lado do usuário (ou client-side).

# Tipos de imagens mapeadas

- Na imagem mapeada **client-side** (no cliente), você inclui todos os detalhes da imagem mapeada no próprio documento.

# Como mapear uma imagem

- Para utilizarmos uma imagem mapeada, precisamos primeiro **definir as áreas** da imagens que irão possuir *links* para outras páginas ou documentos e posteriormente colocar a referida imagem no website. Para realizar este procedimento, podemos definir as áreas utilizando **formas geométricas**, como **quadrados, retângulos, círculos** ou qualquer outro **polígono**.

# Como mapear uma imagem

- Para delinear as formas geométricas na imagens, são utilizadas **coordenadas X e Y**, que são contadas **pixel a pixel**, tendo como 0 (zero) o ponto esquerdo superior da imagem. Para cada tipo de **forma**, a inserção de **coordenadas** possui uma **sintaxe** única.

# Formas geométricas

- Retângulo  
coords="x1,y1,x2,y2"  
(X1 e Y1 são o vértice esquerdo superior e X2 Y2 são o vértice direito inferior)
- Círculo  
coords="x,y,r"  
(X, Y são o centro do círculo e R é a medida do raio, em pixels)
- Polígono  
coords="x1,y1,x2,y2,x3,y3,..."  
(Todos os vértices do polígono)

# Mapeando uma imagem

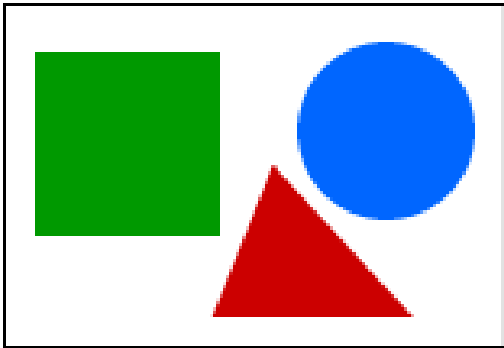
- É muito importante lembrar que atualmente quase nenhuma pessoa realiza a operação de descobrimento dos **pontos de coordenadas** manualmente.

# Mapeando uma imagem

- Praticamente todos os programas **editores de HTML** possuem hoje um **mapeador de imagens** incluso, no qual você desenha as formas geométricas e o programa fornece o **código** pronto. É um processo demorado e trabalhoso realizar tudo manualmente. Entretanto, é de extrema importância que se conheça os conceitos de como o código é construído.

# A Tag MAP

```
  
<map name="mapa_figuras">  
  <area shape="retangulo" coords="9,15,68,73" href="retangulo.html">  
  <area shape="circle" coords="119,40,30" href="circulo.html">  
  <area shape="poligono" coords="83,49,127,98,64,98" href="poligono.html">  
</map>
```





---

# **CRIANDO UMA TABELA**

---

# A tag Table

- Vamos iniciar criando uma **tabela** simples. Depois iremos observar algumas características, tais como:
  - **título**,
  - **borda** e
  - **largura**.
- Uma tabela é constituída de **linhas** e **colunas**, que podem ou não conter dados. Para sua criação, é utilizada a tag **TABLE** `<table>` com a tag de fechamento `</table>`, e as tags para criar linhas, colunas, título de linhas e títulos de colunas, são incluídas entre o início e fim da tag **TABLE**.

# Table Row

- Para **adicionar linhas** em uma tabela, utilizamos a *tag Table Row* **<TR>**, que é inserida entre o de início e fim da tag TABLE. TR é uma tag que **não requer fechamento**. Observe o exemplo na imagem.

# Table Data

- Utilizaremos a tag ***Table Data*** <TD> para inserir dados numa tabela. Diferentemente da *tag* <TR>, esta precisa de **fechamento**, e é utilizada também para controlar o **número de colunas** de sua tabela.

# Table Data

- É fácil imaginar este processo: depois de criada a linha, vamos inserir dados utilizando a *tag* TD. **Para cada conjunto de dados TD, uma célula é criada.** Se colocamos três TDs dentro de uma linha, eles representarão três células. Logo, sua tabela irá apresentar três colunas, uma para cada célula. **Observe a imagem** para ter uma melhor ideia de como este processo funciona.

# Table Data

```
<table>
<tr>
<td>Linha 1</td>
<td>Linha 1</td>
<td>Linha 1</td>
</tr>
<tr>
<td>Linha 2</td>
<td>Linha 2</td>
<td>Linha 2</td>
</tr>
<tr>
<td>Linha 3</td>
<td>Linha 3</td>
<td>Linha 3</td>
</tr>
</table>
```

Linha 1	Linha 1	Linha 1
Linha 2	Linha 2	Linha 2
Linha 3	Linha 3	Linha 3

# Colocando uma borda na tabela

- Podemos utilizar **BORDER** e **WIDTH**, para tratar respectivamente a **borda** e a **largura** da tabela. **Sua altura é automaticamente calculada** pela quantidade de linhas.
- Com *BORDER*, utilizado na *tag TABLE*, podemos adicionar uma borda a uma tabela, **especificando sua largura em pixels.**

# Colocando uma borda na tabela

```
<table border="1px">
<tr>
<td>Linha 1</td>
<td>Linha 1</td>
<td>Linha 1</td>
</tr>
<tr>
<td>Linha 2</td>
<td>Linha 2</td>
<td>Linha 2</td>
</tr>
<tr>
<td>Linha 3</td>
<td>Linha 3</td>
<td>Linha 3</td>
</tr>
</table>
```

Linha 1	Linha 1	Linha 1
Linha 2	Linha 2	Linha 2
Linha 3	Linha 3	Linha 3



# Largura da tabela

- O atributo **WIDTH** é utilizado na *tag* **TABLE** para ajustar a **largura relativa ou absoluta** de uma tabela na janela do browser.
- Os valores podem ser especificados tanto em **porcentagem** quanto em **valores absolutos**. Com os valores absolutos, você deve incluir também um **sufixo** que representa a unidade utilizada.

# Largura da tabela

```
<table border="1px" width="100%">
<tr>
<td>Linha 1</td>
<td>Linha 1</td>
<td>Linha 1</td>
</tr>
<tr>
<td>Linha 2</td>
<td>Linha 2</td>
<td>Linha 2</td>
</tr>
<tr>
<td>Linha 3</td>
<td>Linha 3</td>
<td>Linha 3</td>
</tr>
</table>
```

Linha 1	Linha 1	Linha 1
Linha 2	Linha 2	Linha 2
Linha 3	Linha 3	Linha 3


# Largura da tabela

- Por exemplo, se você quer especificar a largura como 3 polegadas, você usa o atributo **WIDTH**, como **WIDTH = "3in"**, no qual o sufixo in representa **polegadas (inches)**.
- Pode-se também utilizar um valor absoluto em **pixels**, como por exemplo 300px.
- Para especificar um valor relativo, utilizamos **porcentagem (%)**. O browser exibirá então a tabela de acordo com o tamanho de sua janela. Se uma tabela é definida com 60% de largura e o usuário **redimensiona** a janela do browser, a tabela será **automaticamente reajustada** para 60% do novo tamanho da janela.

# Alinhando a tabela

- Utilizaremos o atributo ***ALIGN***, na tag ***TABLE***, para **alinhar** uma tabela **em relação à janela do browser**. Você pode especificar os valores para o atributo: ***LEFT*** (esquerda), ***RIGHT*** (direita) ou ***CENTER*** (centralizada). A sintaxe do ***ALIGN*** é exibida abaixo.

```
<table border="1px" width="100%" align="">  
<tr>  
<td>Linha 1</td>  
<td>Linha 1</td>
```



A dropdown menu is shown next to the 'align' attribute in the code snippet. It contains three options: 'center', 'left', and 'right', each preceded by a small icon representing the alignment (center, left, and right respectively).

- center
- left
- right

---

# FORMATANDO UMA TABELA

---

# Organizando os dados de sua tabela

- Neste módulo, você aprenderá sobre tags e atributos usados para **adicionar funcionalidades** a uma tabela de uma página Web.
- Também serão abordadas as tags utilizadas para melhorar a **aparência** de uma tabela. Para especificar o espaço entre as células e seu conteúdo numa tabela, são utilizados dois atributos com a tag *TABLE*.

# Cellspacing e Cellpadding

- O atributo ***CELLSPACING***, da tag *TABLE*, especifica o **espaço entre a borda da tabela e as células**, sempre em pixels.

```
<table border="1px" cellspacing="20" width="50%">
<tr>
<td>Linha 1</td>
<td>Linha 1</td>
<td>Linha 1</td>
</tr>
<tr>
<td>Linha 2</td>
<td>Linha 2</td>
<td>Linha 2</td>
</tr>
<tr>
<td>Linha 3</td>
<td>Linha 3</td>
<td>Linha 3</td>
</tr>
</table>
```

Linha 1	Linha 1	Linha 1
Linha 2	Linha 2	Linha 2
Linha 3	Linha 3	Linha 3

# Cellspacing e Cellpadding

- Já o atributo ***CELLPADDING***, também da tag ***TABLE***, especifica o **espaço em pixels entre os dados e a borda de uma célula**.

```
<table border="1px" cellpadding="20" width="50%">
<tr>
<td>Linha 1</td>
<td>Linha 1</td>
<td>Linha 1</td>
</tr>
<tr>
<td>Linha 2</td>
<td>Linha 2</td>
<td>Linha 2</td>
</tr>
<tr>
<td>Linha 3</td>
<td>Linha 3</td>
<td>Linha 3</td>
</tr>
</table>
```

Linha 1	Linha 1	Linha 1
Linha 2	Linha 2	Linha 2
Linha 3	Linha 3	Linha 3



# Alinhando dados horizontalmente

- Vamos agora analisar os **atributos** da *tag* **TD**. Podemos mudar as **propriedades das células** de uma tabela utilizando alguns atributos como:
  - ALIGN
  - VALIGN
  - COLSPAN
  - ROWSPAN

# Alinhando dados horizontalmente

- O atributo ***ALIGN*** alinha horizontalmente os **dados** de uma célula. Seu **valor** pode ser especificado como: ***LEFT*** (esquerda), ***RIGHT*** (direita) ou ***CENTER*** (centro).
- O valor left (esquerda) é utilizado para alinhar os dados com a borda esquerda da célula; o valor RIGHT (direita), para alinhá-los com a borda direita da célula, e o valor CENTER para alinhar os dados ao centro.

# Alinhando dados verticalmente

- Podemos utilizar o atributo **VALIGN** para **alinhar** os dados de uma célula **verticalmente** e especificar o valor para este atributo como **TOP** (topo), **BOTTOM** (abaixo). **MIDDLE** (meio) ou **CENTER** (centro).

```
<table border="1px">
<tr>
<td valign="bottom" height="200">Linha 1</td>
<td valign="middle">Linha 1</td>
<td valign="top">Linha 1</td>
</tr>
</table>
```

		Linha 1
	Linha 1	
Linha 1		

# Expandindo uma linha

- O atributo **ROWSPAN** é utilizado para expansão vertical, a fim de que a célula ocupe mais de uma linha. O valor para este atributo deve ser especificado em números.

```
<table border="1px" cellpadding="20" width="50%">
<tr>
<td>Coluna 1</td>
<td>Coluna 2</td>
<td>Coluna 3</td>
<td>Coluna 4</td>
</tr>
<tr>
<td>Dados 1</td>
<td>Dados 2</td>
<td rowspan="3">Dados 3</td>
</tr>
</table>
```

Coluna 1	Coluna 2	Coluna 3	Coluna 4
Dados 1	Dados 2	Dados 3	

# Expandindo uma coluna

- O atributo **COLSPAN** é utilizado para que a célula ocupe mais de uma coluna. O **valor** para este atributo deve ser especificado em **números**.

```
<table border="1px" cellpadding="20" width="50%">
<tr>
<td colspan="4">Coluna 1</td>
</tr>
<tr>
<td>Dados 1</td>
<td>Dados 2</td>
<td>Dados 3</td>
<td>Dados 4</td>
</tr>
</table>
```

Coluna 1			
Dados 1	Dados 2	Dados 3	Dados 4

---

# **CRIANDO FORMULÁRIOS E CAMPOS DE ENTRADA**

---

# Formulários e campos de entrada

- Neste módulo, veremos como criar formulários em uma página Web e também a adicionar campos de entrada de dados em um formulário.

# O que é um formulário?

- Um formulário é um questionário formatado, com campos a serem preenchidos.
- Em uma página Web, utilizamos um **formulário** para **receber informações** ou respostas dos usuários em forma de **texto**, **lista de opções**, **múltipla escolha** ou **escolha única**.



# O que é um formulário?

- Por exemplo, podemos usar um formulário para receber os detalhes da **compra** de um **cliente**.
- O browser processa as entradas do formulário enviando-as ao servidor Web responsável.
- **Programas** armazenados no servidor processam a informação e o browser exibe o resultado em sua janela.

# A tag Form

- No HTML, a tag *FORM* cria um formulário.

`<form></form>`

- Sempre que existir a necessidade de incluir um formulário numa página web devemos inserí-lo iniciando com a tag `<form>` e encerrá-lo utilizando a tag `</form>`.
- Entre estas duas tags colocaremos as tags que criam campos de entrada, formatam o texto e fornecem instruções sobre o que fazer com os dados que são inseridos.

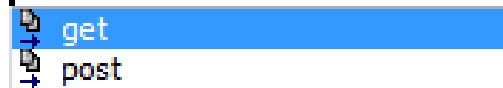
# O atributo Method

- O atributo ***METHOD*** especifica a maneira com que o browser deve **enviar os dados** do formulário para o servidor web, para os dados serem processados. **METHOD** pode possuir dois valores:
  - **POST**: o browser **enviará diretamente** o fluxo de dados para o servidor Web.
  - **GET**: o fluxo de dados será **anexado a URL do *script*** e enviado ao servidor como uma única *URL*.

# O atributo Action

- O segundo atributo utilizado com a tag *FORM* é o ***ACTION***. Ele **especifica a URL do script** que irá processar os dados do formulário.

```
<form action="URL" method="">
```



```
</form>
```

# Tipos de Entrada

- Em um formulário, podemos receber diferentes tipos de entrada dos usuários, tais como:
  - Texto;
  - Senha;
  - Uma única opção de um conjunto de opções;
  - Várias opções de um conjunto de opções.

# A tag INPUT e seus atributos

- Podemos utilizar diferentes tags para criar vários tipos de **campos de entrada** em um **formulário**, como:
  - Caixa de texto,
  - Campo de senha,
  - Campos ocultos,
  - *Checkboxes* (múltipla escolha),
  - *Radio buttons* (botões de escolha única).

# A tag INPUT e seus atributos

- Vamos agora analisar os atributos da *tag* **<INPUT>**:
  - **TYPE**: especifica o **tipo de entrada** que você precisa receber do usuário.
  - **NAME**: é utilizado para especificar a **variável que armazena o valor** que o usuário digita no campo de entrada. O script do lado do servidor usa esta variável durante o **processamento dos dados**.
  - **ID**: especifica um identificador único para o campo.

# Os possíveis valores de Type

- O atributo ***TYPE***, especifica o **tipo de controle** (ou tipo de campo) que será criado.
  - **text**: cria um campo de entrada de texto de uma linha.
  - **password**: cria um campo de entrada de texto sem exibir o que está sendo digitado.
  - **radio**: cria um botão de escolha única.
  - **file**: cria um controle para seleção de arquivo. O valor do atributo deve ser configurado como o nome do arquivo.
  - **checkbox**: cria uma caixa de seleção.



# Os possíveis valores de Type

- **reset**: cria um botão de reset, que limpa todos os campos já preenchidos.
- **image**: cria um botão de enviar baseado numa imagem específica, cuja URL deve ser fornecida.
- **button**: cria um botão para ser pressionado. O valor do atributo deve ser configurado com o mesmo nome do botão.
- **submit**: cria um botão de enviar.
- **hidden**: cria um controle oculto.

# Atributos adicionais

- Além desses atributos, você usa alguns atributos específicos com a tag ***INPUT***, dependendo do tipo de campo de entrada que você quer adicionar ao formulário.
  - **type**: Especifica o tipo de controle a ser criado. Se nada for especificado, o padrão é "text".
  - **maxlength**: Quando o valor de TYPE for "text" ou "password", este especifica o número máximo de caracteres do campo.

# Atributos adicionais

- **src:** quando o valor de TYPE for "image", este atributo especifica a localização da imagem usada para decorar o botão de enviar.
- **checked:** quando TYPE tem o valor "radio" ou "checkbox", este atributo significa que o botão está "selecionado", "marcado", "ligado".
- **value:** especifica o valor inicial do controle. É opcional, com exceção de quando o valor do atributo TYPE é "radio" ou "checkbox".
- **name:** atribui o nome do controle.
- **size:** especifica o tamanho inicial do controle. a largura é dada em pixels, com exceção de quando o valor for "text" ou "password". Nestes casos o valor se refere ao número de caracteres.

# A sintaxe da tag Input

- Você usa uma caixa de texto em um formulário para aceitar uma **única linha de texto** do usuário.

```
<input type="text" name="txt_cpf" size="11" maxlength="11" />
```



Valor utilizado para especificar o tipo de entrada como texto.



Valor para especificar a variável que armazena os dados informados .



Tamanho do campo, em número de caracteres.



Número máximo de caracteres que podem ser inseridos pelo usuário.

# Inserindo um campo de texto

- Por exemplo, se você precisar adicionar uma caixa de texto para receber o CPF de um cliente num formulário de pedido online:

```
<table border="1px" cellpadding="5" width="250">
<tr>
<td align="right">Informe o CPF:</td>
<td><input type="text" name="txt_cpf" size="11" maxlength="11" /></td>
</tr>
</table>
```

Informe o CPF:	<input type="text"/>
----------------	----------------------

# Inserindo um campo de texto

- Por **padrão**, os browsers exibem uma **caixa de texto vazia** no formulário. Caso exista a **necessidade de exibir** um texto padrão, pode-se usar o atributo **VALUE** da *tag INPUT*, como exemplificado abaixo:

```
<table border="1px" cellpadding="5" width="250">
<tr>
<td align="right">Nome:</td>
<td><input type="text" name="txt_cpf" size="50" maxlength="50" value="Informe o seu nome" /></td>
</tr>
</table>
```

Nome:	Informe o seu nome
-------	--------------------

# Inserindo um campo de senha

- Agora, veremos como adicionar um **campo de senha** em um formulário. Este tipo de campo é utilizado para que as **informações** inseridas **não sejam exibidas na tela**.

```
<table border="1px" cellpadding="5" width="250">
<tr>
<td align="right">Senha:</td>
<td><input type="password" name="txt_senha" size="12" maxlength="12" /></td>
</tr>
<tr>
<td align="right">Confirme a Senha:</td>
<td><input type="password" name="txt_confirme_senha" size="12" maxlength="12" /></td>
</tr>
</table>
```

Senha:	<input type="password"/>
Confirme a Senha:	<input type="password"/>

# Inserindo um campo oculto

- Vamos verificar agora como adicionar um campo oculto em um formulário para enviar uma palavra chave, um número de validação ou qualquer string *para o servidor*.

```
<input type="hidden" name="NOME_VARIAVEL" value="VALOR_VARIAVEL" />
```

- Este campo não será exibido no formulário, porém, pode possuir informações muito importantes para o servidor, como um valor ou palavra chave que valide os números de cartão de crédito.



# Inserindo botões de escolha única

- ***Radio buttons*** são utilizados para permitir ao usuário selecionar **uma opção única dentre um conjunto de opções** exibidas em um formulário.

```
<table border="1px" cellpadding="5" width="250">
  <tr>
    <td align="right">Sexo:</td>
    <td><input type="radio" name="sexo" value="masculino">
      Masculino
      <input type="radio" name="sexo" value="feminino">
      Feminino </td>
  </tr>
</table>
```

Sexo:	<input type="radio"/> Masculino <input type="radio"/> Feminino
-------	--

# Inserindo botões de escolha única

- Quando você exibe um conjunto de opções em um formulário, pode querer **exibir uma das opções selecionada** por padrão. Para fazer isto, você usa o atributo ***CHECKED*** da tag ***INPUT***.

```
<table border="1px" cellpadding="5" width="250">
  <tr>
    <td align="right">Sexo:</td>
    <td><input type="radio" name="sexo" value="masculino" checked="checked">
      Masculino
      <input type="radio" name="sexo" value="feminino">
      Feminino </td>
  </tr>
</table>
```

Sexo:	<input checked="" type="radio"/> Masculino	<input type="radio"/> Feminino
-------	--	--------------------------------

# Inserindo botões de múltipla escolha

- Agora veremos como adicionar um conjunto de **checkboxes** (botões de múltipla escolha) para permitir ao usuário **selecionar várias opções**.

```
<table border="1px" cellpadding="5" width="250">
  <tr>
    <td align="right">Assuntos de interesse:</td>
    <td><input type="checkbox" name="interesse" value="esporte" />Esportes <br />
      <input type="checkbox" name="interesse" value="culinaria" />Culinária <br />
      <input type="checkbox" name="interesse" value="economia" />Economia </td>
  </tr>
</table>
```

Assuntos de interesse:	<input type="checkbox"/> Esportes <input type="checkbox"/> Culinária <input type="checkbox"/> Economia
------------------------	--

# Inserindo botões de múltipla escolha

- Assim como nos botões *radio*, podemos escolher uma ou mais opções para aparecerem **selecionadas por padrão**. Para isto basta adicionar a palavra **CHECKED** dentro da *tag INPUT* desejada.

```
<table border="1px" cellpadding="5" width="250">
  <tr>
    <td align="right">Assuntos de interesse:</td>
    <td><input type="checkbox" name="interesse" value="esporte" />Esportes <br />
      <input type="checkbox" name="interesse" value="culinaria" />Culinária <br />
      <input type="checkbox" name="interesse" value="outros" checked="checked" />Outros </td>
  </tr>
</table>
```

Assuntos de interesse:	<input type="checkbox"/> Esportes <input type="checkbox"/> Culinária <input checked="" type="checkbox"/> Outros
------------------------	---

# Inserindo uma área de texto

- Veremos como adicionar uma área de texto em um formulário, para receber várias linhas de texto de um usuário. A *tag* **TEXTAREA** adiciona uma **área de texto** em um formulário.

# Inserindo uma área de texto

- O atributo **ROWS** especifica o número de linhas da área de texto.
- Já o atributo **COLS** serve para especificar a largura da área de texto especificando o numero de colunas.

```
<table border="1px" cellpadding="5" width="250">  
  <tr>  
    <td align="right">Mensagem:</td>  
    <td><textarea name="mensagem" rows="5" cols="40"></textarea></td>  
  </tr>  
</table>
```

---

# **ADICIONANDO BOTÕES E MENUS A UM FORMULÁRIO**

---

# Como criar botões e menus

- Neste módulo, veremos como adicionar diferentes tipos de botões e menus em um formulário.



# O botão reset

- O botão **Reset** limpa as entradas que foram feitas em um formulário, ou seja, **apaga as informações** que já foram inseridas em todos os campos.

```
<input type="reset" value="Limpar" />
```



Tag que cria botões ou campos de entrada.



Tipo de botão ou campo a ser criado



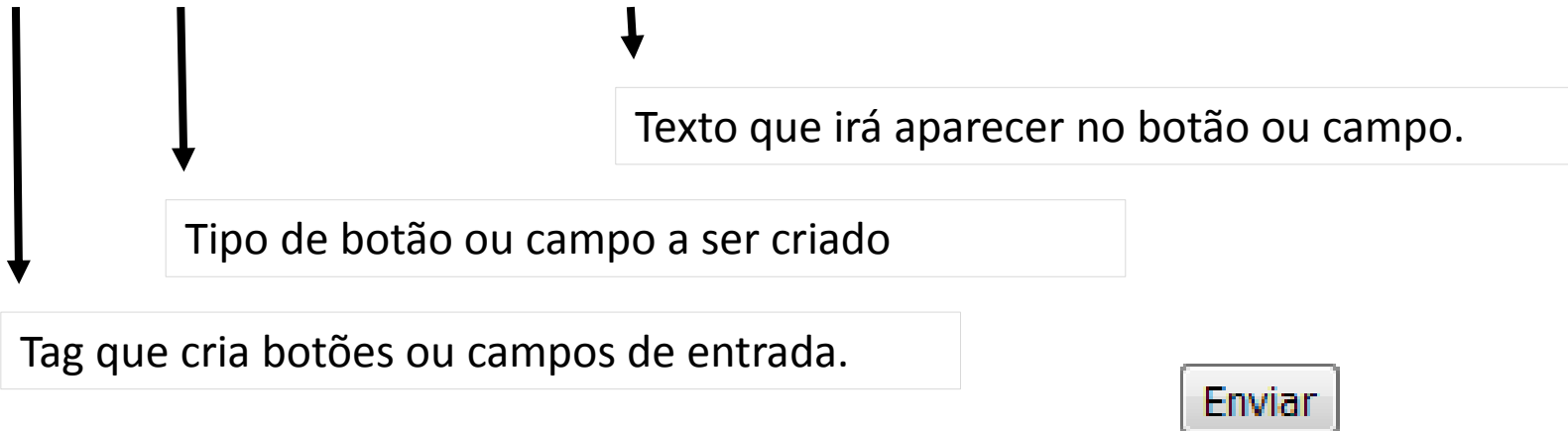
Texto que irá aparecer no botão ou campo.

Limpar

# O botão submit

- O botão ***Submit***, em um formulário, **envia ao servidor Web** a informação digitada pelo usuário para ser processada.

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



# Tipos de menus

- Vamos agora analisar como adicionar **menus** em um formulário para permitir ao usuário **selecionar uma ou mais opções**, a partir de um conjunto de opções.

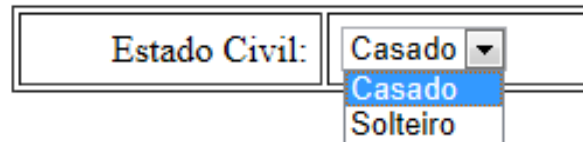
# Tipos de menus

- Os menus em um formulário podem ser de **dois tipos**:
  - Menu ***Drop Down***;
  - Menu de **Escolhas** (uma ou múltiplas)

# O menu drop-down

- Um menu **Drop Down** permite ao usuário **selecionar uma única opção** de um conjunto de opções pré-determinadas. A **tag *SELECT*** cria um menu Drop-down.

```
<table border="1px" cellpadding="5" width="250">
  <tr>
    <td align="right">Estado Civil:</td>
    <td><select name="estado_civil">
      <option value="casado">Casado</option>
      <option value="solteiro">Solteiro</option>
    </select></td>
  </tr>
</table>
```



Visual representation of the HTML code above, showing a form with a label "Estado Civil:" and a dropdown menu with options "Casado" and "Solteiro". The dropdown menu is currently open, showing the options.

# O menu drop down com visualização diferenciada

- O atributo ***SELECTED*** especifica o item que **será exibido por padrão**. O atributo ***SIZE*** controla a aparência da caixa de seleção numa página *Web*, já que ele **determina o número de itens que será exibido** no menu.

```
<table border="1px" cellpadding="5" width="250">
  <tr>
    <td align="right">Estado Civil:</td>
    <td><select name="estado_civil" size="2">
      <option value="casado">Casado</option>
      <option value="solteiro">Solteiro</option>
    </select></td>
  </tr>
</table>
```

Estado Civil:	<div>Casado Solteiro</div>
---------------	--------------------------------

# Selecionando múltiplas opções de um menu

- Agora vamos verificar como inserir um menu *scrolling* em um formulário. Este tipo de menu permite ao usuário selecionar uma ou mais opções de um conjunto de opções dado. A mesma *tag* *SELECT* cria um menu *scrolling*. Para isso devemos utilizar o atributo *MULTIPLE* para permitir ao usuário selecionar mais de uma opção do menu.

# Selecionando múltiplas opções de um menu

```
<table border="1px" cellpadding="5" width="400">
  <tr>
    <td align="right">Selecione os assuntos de Interesse:</td>
    <td><select name="assuntos_interesse" multiple="multiple">
      <option value="esportes">Esportes</option>
      <option value="economia">Economia</option>
      <option value="lazer">Lazer</option>
    </select></td>
  </tr>
</table>
```

Selecione os assuntos de Interesse:	<div style="border: 1px solid black; padding: 2px;"><div style="background-color: #007bff; color: white; padding: 2px;">Esportes</div><div style="background-color: #007bff; color: white; padding: 2px;">Economia</div><div style="padding: 2px;">Lazer</div></div>
-------------------------------------	--



---

# **ATRIBUTOS DAS TAGS BODY E TABLE**

---

# Extensões para as tags body e table

- Neste módulo vamos analisar algumas tags HTML que podem ajudar na **formatação** e **layout** de **tabelas** e outros elementos. Falaremos sobre as tags ***BODY*** e ***TABLE***.

# Melhorando a aparência do fundo de uma página Web

- O *HTML* fornece o atributo ***BACKGROUND*** para a *tag* ***BODY***. Com ele você pode exibir uma **imagem no fundo de uma página Web**. Observe o código e seu resultado no navegador.

# Alterando a cor de fundo de uma tabela

- Além da tag *BODY*, você poderá aplicar o atributo *BGCOLOR* dentro de tabelas, ou seja, dentro das tags `<td>` e `<tr>`.
- Verifique abaixo a sintaxe do atributo ***BGCOLOR***. O código ou o nome da cor devem ser especificados como o valor do atributo, sempre utilizando códigos hexadecimais. Repare no código colocado abaixo e a tabela gerada a partir dele, onde colocamos uma cor de fundo nas duas linhas.

---

# A TAG META

---

# A tag META

- Metadados são **informações** sobre seus dados. Neste caso, a tag <META> fornece dados sobre seu documento HTML.
- Estes dados, no entanto, não são exibidos para o leitor ou usuário final. Eles servem para **descrever**, geralmente, **definições** sobre a página, **palavras-chave**, informações sobre o **autor**, quando foi a **última modificação**, e outras informações.
  - Vale lembrar que a tag <meta> deve ser inserida dentro do cabeçalho da página, ou seja, entre as tags <head> e </head>.

# A tag META

- Todos estes dados podem ser utilizados por **browsers** (informações sobre como exibir o conteúdo ou recarregar a página), **sistemas de busca**, e outros serviços.
- Vamos agora verificar como colocar estas informações na página:

# O atributo **CONTENT**

- O atributo **CONTENT** possui somente texto, e especifica o **conteúdo** da informação meta.
  - O valor de CONTENT depende do valor do atributo name ou do HTTP-equiv.

```
<meta name="author" content="João da Silva">
```

Neste exemplo o content="João da Silva" se refere ao nome do autor da página, especificado como name="author".

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

Neste exemplo é definido o tipo de conteúdo da página e o tipo de codificação.



# O atributo **NAME**

- **NAME** - Fornece um **nome** para a informação no atributo **CONTENT**.
  - **Author** - define o autor do documento;
  - **Description** – define a descrição do documento;
  - **Keywords**- define as palavras-chave que descrevem o documento;
  - **Generator** – define o programa utilizado para gerar o documento;
  - **Robots** – define regras para os robôs (programas utilizados em sistemas de busca);
  - **Copyright** – define informações de direitos autorais do documento.

# O atributo NAME

```
<head>
<meta name="author" content="João da Silva">
<meta name="description" content="Página criada como exemplo">
<meta name="keywords" content="Faculdade Pitágoras, Aula, Desenvolvimento Web">
<meta name="robots" content="João da Silva">
<meta Name="robots" content="index">
<meta name="copyright" content="2012@ http://www.faculdadepitagoras.com.br" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Desenvolvimento Web</title>
</head>
```

# O atributo HTTP-EQUIV

- **HTTP\_EQUIV** - Fornece um cabeçalho HTTP para a informação no atributo *CONTENT*. Seu valor depende do valor de *CONTENT*. Caso utilize *NAME*, não deve utilizar HTTP-equiv.
  - **content-type** – Define o tipo MIME do document (como text/html);
  - **expires** – define quando o documento será considerado obsoleto;
  - **refresh** – define o intervalo de tempo para que o documento recarregue;
  - **set-cookie** – define o valor de um cookie.

# O atributo HTTP-EQUIV

- A sintaxe da *tag META* para **recarregar** uma página **automaticamente** é:

```
<meta http-equiv="refresh" content="SEGUNDOS" url="URL">
```

# Linguagem de Programação

- Uma linguagem de programação é um vocabulário e um conjunto de regras gramaticais usadas para escrever programas de computador.
- Esses programas instruem o computador a realizar determinadas tarefas específicas.
- Cada linguagem possui um conjunto único de palavras-chaves (palavras que ela reconhece) e uma sintaxe (regras) específica para organizar as instruções dos programas.



# ASCII

- *American Standard Code for Information Interchange* (Código Padrão Americano para o Intercâmbio de Informação) é uma codificação de caracteres em oito bits, baseada no alfabeto norte-americano. Todos os caracteres são convertidos em bits utilizando esta tabela.

