HTML para Concursos Públicos

Aula 05 – Formulários

➢O que são Formulários?

Os Formulários são elementos que permitem o envio de dados de uma página HTML para o servidor Web.

Desta forma, os Formulários permitem a criação de páginas que sejam mais interativas, onde o usuário final consiga trocar informações com um servidor Web. Geralmente quem trabalha com Formulários, também trabalha com *scripts* do lado servidor, tais como PHP, ASP, JSP etc.

NESTE AULA, iremos estudar os elementos (as tags) de Formulário referentes à versão 4.01 da HTML. Na versão 5, novos elementos e atributos foram criados, mas isto ficará para um módulo HTML posterior.

➢Estrutura de um Formulário

Um Formulário é composto pela tag <form> e outros elementos de entrada de dados, como caixas de texto, listas de seleção, caixas de seleção, botões etc.

```
<form action="login.php" method="post">
  Usuário:<input type="text"><br>
  Senha:<input type="password"><br>
  <input type="submit" value="Entrar">
</form>
```

No arquivo HTML...

...e no *browser*.

Neste exemplo, temos a *tag* <form> iniciando o formulário, e a *tag* </form> finalizando-o. Por isso, diz-se que o elemento <form> não pode ser vazio.

Dentro das *tags* <form> e </form> temos texto comum acompanhado das *tags* <input>. Estas *tags* definem alguns dos elementos de interação (caixas de texto e botões).

Usamos a *tag*
 apenas para quebrar o fluxo de texto entre os elementos <input>, pois estas *tags* são Elementos de Linha, ou seja, não quebram linhas antes ou depois.

➢<form> e seus atributos principais

A tag <form>, que define um formulário, possui atributos importantes, que determinam como os dados do formulário devem ser tratados.

```
<form action="login.php" method="post“>
  Usuário:<input type="text"><br>
  Senha:<input type="password"><br>
  <input type="submit" value="Entrar">
</form>
```

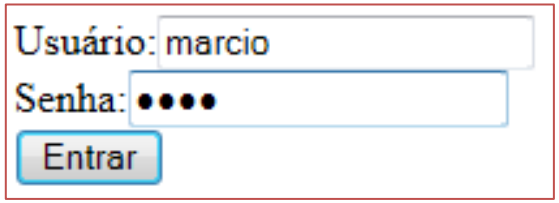
O atributo **action** define qual página deverá ser chamada para receber as informações do formulário e processá-las. Geralmente aponta para um arquivo que contem *scripts* de servidor. No nosso exemplo, aponta para o arquivo “login.php”.

O atributo **method** define qual método deve ser usado para enviar as informações para o servidor web. Existem dois métodos: GET e POST, que veremos a seguir.

➢Método GET

O Método GET envia as informações do formulário através da URL em pares de "Nomes" e "Valores". Por exemplo:

```
<form action="login.php" method="get">
  Usuário:<input name="usuario" type="text"><br>
  Senha:<input name="senha" type="password"><br>
  <input type="submit" value="Entrar">
</form>
```



Ao clicarmos no botão, seremos direcionados para a seguinte página:
login.php?usuario=marcio&senha=1234
Onde 1234 foi o texto que entramos para a segunda caixa de texto acima.

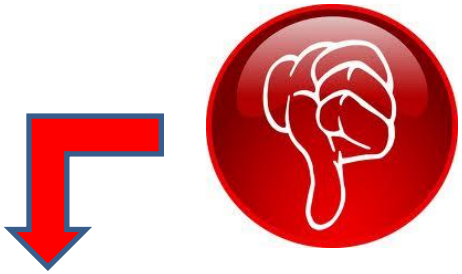
O *browser* insere o caracter "?" para indicar que o que vêm depois deste símbolo são os pares de valores "nome" e "valor" dos elementos *inputs* do formulário. O Nome do *input* é dado pelo atributo "*name*". Para separar os pares, é usado o símbolo "&".

É o método PADRÃO de envio de dados. Ou seja, se não for especificado no atributo "method" da tag <form> então, assume-se que o método é o GET.

Quando usar o método GET

É muito útil quando queremos fazer um “bookmark” para a página, pois os valores dos elementos de entrada são passados JUNTOS com a URL da página.

MAS....



Não deve ser usado quando se deseja enviar dados sigilosos, como senhas (no nosso exemplo). Pois as informações ficarão visíveis na URL

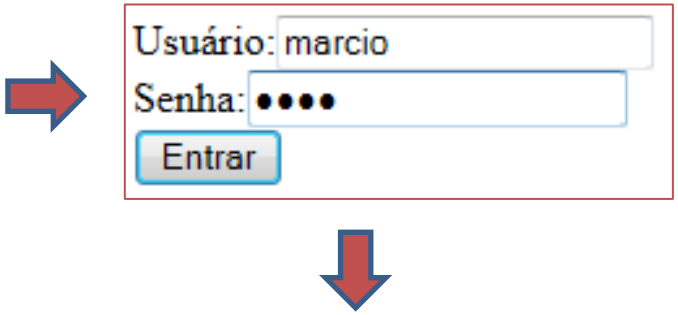
Existe a limitação de 3000 caracteres para envio de informações, pois estas vão com a URL no *browser*.

Alguns dados, tais como arquivos, não podem ser enviados pelo método GET.

➢Método POST

O Método POST envia as informações do formulário dentro da requisição HTTP. Por exemplo:

```
<form action="login.php" method="post">
  Usuário:<input name="usuario" type="text"><br>
  Senha:<input name="senha" type="password"><br>
  <input type="submit" value="Entrar">
</form>
```



Ao clicarmos no botão, seremos direcionados para a página:*login.php.* Os textos "marcio" e "1234" serão encaminhados juntamente com a requisição HTTP, não ficando visíveis na URL. Isso supondo que foi entrado o texto "1234" na segunda caixa de texto.

É um método mais seguro para passar informações, já que os dados não ficam visíveis.

Além disso, não há limitações de quanta informação pode ser passada. Exceto se o servidor web contiver alguma restrição.

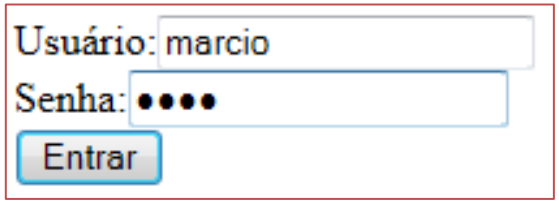
A única limitação que existe neste método, é que não podemos fazer um bookmark para a página, pois as informações não são passadas na URL.

➢Validações de Formulários

Provas de TI
Tecnologia da Informação para Concursos

Às vezes, precisamos validar as informações que são entradas nos formulários, para evitar que dados errados ou mal intencionados possam ser enviados.

```
<form action="login.php" method="post" onsubmit=
"return validarEntradas(this);">
  Usuário:<input name="usuario" type="text"><br>
  Senha:<input name="senha" type="password"><br>
  <input type="submit" value="Entrar">
</form>
```



Ao clicarmos no botão, a rotina escrita em *javascript* "`validarEntradas(this)`" será chamada para validar as entradas.

Isto ocorre, pois o atributo "`onsubmit`" da tag <`form`> especifica uma rotina de script no lado cliente (geralmente *javascript*) que deve ser executada ANTES de enviar (*submit*) os dados para o servidor.

Se a rotina *javascript* retornar `TRUE`, é porque o formulário pode ser enviado ao servidor web. Mas se retornar `FALSE`, o formulário NÃO é enviado.

➢O elemento <input>

O elemento <input> é onde ficam armazenadas as informações que queremos enviar ao servidor web. Possui vários atributos que definem o seu comportamento.

`type` : especifica o tipo de *input* que queremos (botão, caixa de texto etc). Pode ter vários valores, que iremos estudar mais a seguir.

`name` : define o nome do *input*. É o que o identifica no momento de enviar as informações para o servidor web.

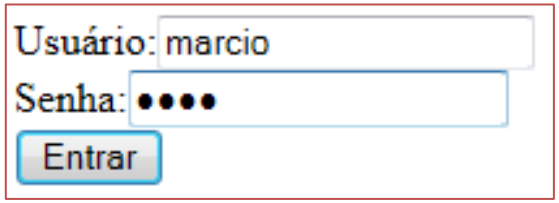
`disabled` : atributo que se estiver presente no input, o desabilita, ou seja, não conseguimos escrever nada nele. Além disso, seu conteúdo não é enviado ao servidor web. Este atributo pode ser modificado em tempo de execução por uma rotina *javascript*.

`readonly`: atributo que se estiver presente no input, deixa seu conteúdo como somente leitura, ou seja, você pode selecionar o conteúdo e copiá-lo, mas não pode alterá-lo. Além disso, seu conteúdo é enviado ao servidor web. Este atributo pode ser modificado em tempo de execução por uma rotina *javascript*.

➢Os vários elementos <input>

Existem vários tipos de elementos <input>, e o comportamento deles é definido baseado no atributo "type".

```
<form action="login.php" method="post">
  Usuário:<input name="usuario" type="text"><br>
  Senha:<input name="senha" type="password"><br>
  <input type="submit" value="Entrar">
</form>
```



No exemplo acima, temos:

type="text" : especifica um campo texto de uma só linha.

type="password" : especifica um campo texto de uma só linha. Seu conteúdo é exibido usando-se um símbolo padrão (máscara).

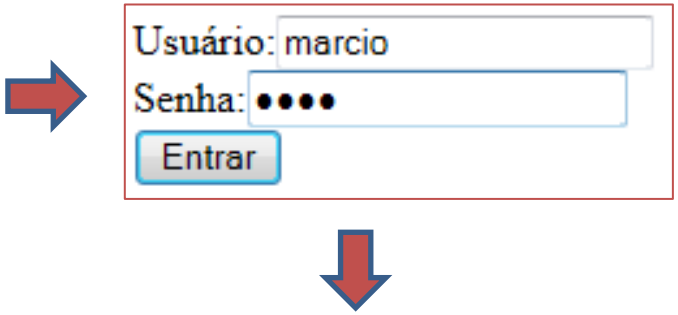
type="submit" : define um botão, que ao ser clicado solicita o envio (*submit*) das informações do formulário para o servidor Web

➢Botão <input>

Provas de TI
Tecnologia da Informação para Concursos

type="button" : define um botão. Geralmente é associada uma rotina *javascript* para ser executada quando clicamos no botão.

```
<form action="login.php" method="post">
  Usuário:<input name="usuario" type="text"><br>
  Senha:<input name="senha" type="password"><br>
  <input type="button" value="Entrar"
    onclick="javascript:iniciarLogin(this);">
</form>
```



No exemplo acima, temos:

type="button" : define o input como um botão.

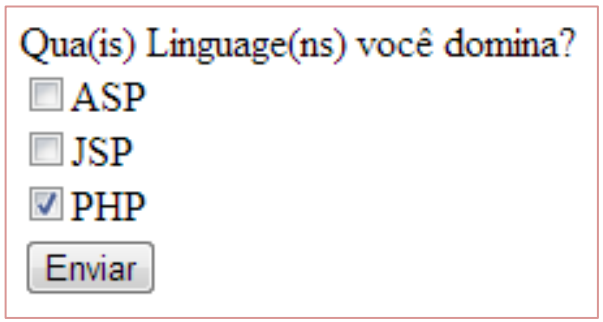
O atributo value define o texto do botão : "Entrar"

E o atributo onclick especifica qual rotina *javascript* será executada ao se clicar no botão: iniciarLogin(this).

➢Caixas de Múltipla Seleção

Através do elemento <input> é possível criar caixas de múltiplas seleções. Assim, o usuário pode selecionar (clicando) entre uma ou várias opções.

```
<form action="selecao.php" method="get">
  Qua(is) Linguage(ns) você domina?<br>
  <input type="checkbox" name="asp"
     value="asp">ASP<br>
  <input type="checkbox" name="jsp"
     value="jsp">JSP<br>
  <input type="checkbox" name="php"
     value="php" checked>PHP<br>
  <input type="submit">
</form>
```



No exemplo acima, temos:

`type="checkbox"` : define o input como uma caixa de seleção.

`value` : define o valor a ser enviado ao servidor Web caso este input esteja selecionado.

`checked`: se estiver no *input*, já deixa esta caixa de seleção como já marcada.

➢Caixas de Seleção Única

Através do elemento <input> é possível criar caixas de seleções únicas. Assim, o usuário só pode selecionar apenas **uma** de várias seleções.

```
<form action="selecao.php" method="post">
  Quanto(s) concurso(s) você já fez?<br>
  <input type="radio" name="qtosConc"
     value="nenhum">Nenhum<br>
  <input type="radio" name=" qtosConc"
     value="SohUm">Só uma vez<br>
  <input type="radio" name=" qtosConc"
     value="varias" checked>Vixe! Um monte.
</form>
```

Quanto(s) concurso(s) você já fez?
- ○ Nenhum
- ○ Só uma vez
- ◉ Vixe! Um monte.

No exemplo acima, temos:

`type="radio"` : define o input como uma caixa de seleção única.

`name`: define o nome do *input*. Este nome "amarra" as caixas de seleção. Assim se o usuário marca uma opção de mesmo nome, desmarca outra de mesmo nome.

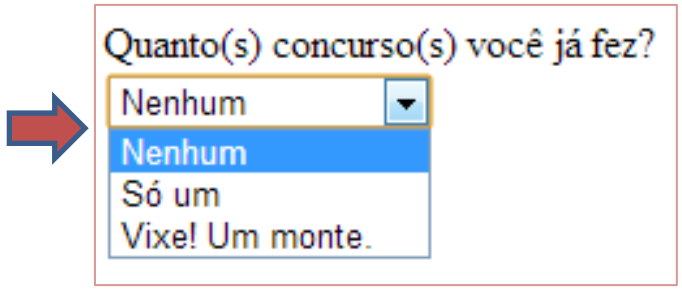
`value` : define o valor a ser enviado ao servidor caso este *input* esteja selecionado.

`checked`: se estiver no *input*, já deixa esta caixa de seleção como já marcada.

➢Lista de Seleção Única

O elemento <select> permite criar LISTAS de seleções únicas. Assim, o usuário só pode selecionar apenas **uma** de várias opções, baseada numa lista de valores.

```
<form action="selecao.php" method="get">
Quanto(s) concurso(s) você já fez?<br>
 <select name="qtosConc">
   <option value="zero">Nenhum</option>
   <option value="um">Só um</option>
   <option value="monte">Vixe! Um monte.</option>
 </select>
</form>
```



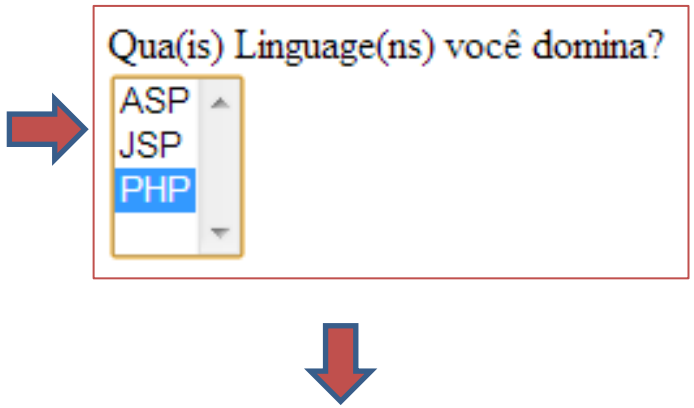
No exemplo acima, temos:

- <select> : define uma lista de seleção única, é o elemento que delimita a lista.
- name: atributo que define o nome do *select*. Este nome é usado para enviar o valor selecionado para o servidor.
- Cada *tag* <option> define o valor a ser enviado ao servidor caso esta opção esteja selecionada na lista.
- selected: se estiver no *option*, já deixa esta opção pré-selecionada.

➢Lista de Seleção Múltipla

O elemento <select> permite criar LISTAS de seleções múltiplas. Assim, o usuário pode selecionar uma ou mais opções, baseada numa lista de valores.

```
<form action="selecao.php" method="get">
  Qua(is) Linguage(ns) você domina?<br>
 <select name="selecao" multiple>
  <option value="asp" value="asp">ASP</option>
  <option value="jsp">JSP</option>
  <option value="php" selected>PHP</option>
 </select>
</form>
```



No exemplo acima, temos:

Atributo `multiple` : define que pode haver várias seleções de elementos na lista.

Cada *tag* <option> define o valor a ser enviado ao servidor caso esta opção esteja selecionada na lista.

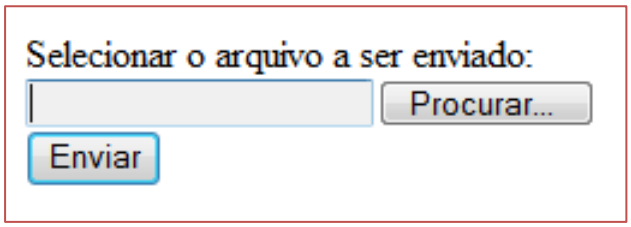
`selected`: se estiver no *option*, já deixa esta opção pré-selecionada.

Recomenda-se usar as Caixas de Seleções Múltiplas (*checkbox*) ao invés desta Lista.

Envio de arquivos

O elemento <input> permite a seleção de arquivos também. E como se fosse uma caixa de texto normal com um botão associado para selecionar arquivo.

```
<form enctype="multipart/form-data"
  action="arquivo.php" method="post">
  Selecionar o arquivo a ser enviado:<br>
  <input type="file" name="arquivo"><br>
  <input type="submit" value="Enviar">
</form>
```



No exemplo acima, temos:

Atributo `enctype` igual a "multipart/form-data": indica que ao enviar as informações, o *browser* irá enviar no formato específico de cada *input*, ou seja, não irá tentar converter para texto. Desta forma, o arquivo será enviado no seu formato.

Para envio neste formato, é SEMPRE necessário usar o método POST.

Resolução de Questões

Questão 1 (FCC - 2011 - TRE-PE - Analista Judiciário - Análise de Sistemas)

Analise o código da página HTML seguinte.
Ao clicar no botão Submit, o formulário não será submetido se a função validar(frm) retornar um valor

a) null.
b) 1.
c) 0.
d) true.
e) false.

```
<html>
  <head>
   <title>Cadastro</title>
    <script type="text/javascript">
     function validar(frm){
     }
    </script>
 <head>
 <body>
  <form name="frm" method="post" action="incluir.jsp"
        onsubmit="return validar(this)">
      <p><label>Nome:<input type="text" name="nome" size="30"></label></p>
      <p><label>Renda: <input type="text" name="renda" size="15"></label></p>
      <p><input type="submit" value="Submit"></p>
  </form>
 </body>
</html>
```

Resposta: Letra E. O *Submit* da página só será realizado se a rotina *javascript* "`validar(this)`" returnar **`true`**. Se for **`false`**, não será submetido.

➢Resolução de Questões

Provas de TI
Tecnologia da Informação para Concursos

Questão 2 (CESGRANRIO - 2011 - Transpetro - Analista de Sistemas Júnior)

Um formulário em HTML precisa ser publicado para que os usuários possam enviar arquivos no formato PDF para serem armazenados em um banco de dados de documentos.
O programador criou em uma página uma *tag* <form> com parâmetros `method="post"` e `action="salva.do"`. Dentro da *tag* <form> foi inserida uma *tag* <input type="file">.
O procedimento está

a) correto, pois basta ter uma *tag* <input type="file"> para que arquivos possam ser submetidos ao servidor.
b) incorreto, pois é preciso também especificar, nesse caso, o atributo `enctype="multipart/form-data"` na tag <form>.
c) incorreto, pois é preciso também especificar, nesse caso, o atributo `enctype="application/x-pdf"` na tag <form>.
d) incorreto, pois é preciso também especificar, nesse caso, o atributo `enctype="application/x-www-form-urlencoded"` na tag <form>.
e) incorreto, pois é preciso também especificar, nesse caso, o atributo `content-type="application/x-pdf"` na tag <form>.

Resposta: Letra B. Além do `input` do tipo "file" e o método POST, é necessário que o `enctype` seja "multipart/form-data".

Resolução de Questões

Provas de TI
Tecnologia da Informação para Concursos

Questão 3 (FCC - 2010 - AL-SP - Agente Legislativo de Serviços Técnicos e Administrativos - Processamento de Dados)

Para impedir que o usuário altere valores em um campo texto de um formulário HTML é correto utilizar

a) o atributo *disabled*, apenas.
b) os atributos *disabled* ou *readonly*.
c) o atributo *readonly*, apenas
d) os atributos *unchecked* ou *disabled*.
e) os atributos *unchecked* ou *readonly*.

Resposta: Letra B. O atributo *disabled* desabilita o input a tal ponto do usuário não conseguir copiar o seu conteúdo. O atributo *readonly* deixa o input acessível, envia para o servidor, mas não deixa alterar o seu conteúdo.

Resolução de Questões

Provas de TI
Tecnologia da Informação para Concursos

Questão 4 (FCC - 2010 - AL-SP - Agente Legislativo de Serviços Técnicos e Administrativos - Processamento de Dados)

GET e POST são alguns dos principais métodos que determinam o que o servidor deve fazer com o URL fornecido no momento da requisição de um recurso. Relacionado a esses métodos, considere:

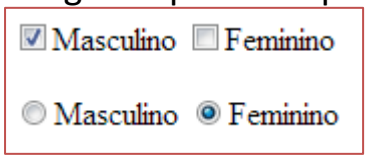
I. Dados enviados em uma requisição utilizando o método GET ficam visíveis na linha de endereço do navegador.
II. Se não for especificado um método, o POST é adotado como padrão.
III. O método GET é geralmente utilizado para enviar grandes quantidades de dados por meio de um formulário.
IV. O método POST não exibe os dados enviados na linha de endereço do navegador.

Está correto o que se afirma APENAS em
a) I e II.
b) I e IV.
c) II, III e IV.
d) III.
e) IV.

Resposta: Letra B. O Método GET é o padrão, e seus dados ficam visíveis na URL (endereço do *browser*). O Método POST deve ser usado para o envio de grandes informações, e seus dados não são exibidos na URL.

Resolução de Questões

Questão 5 (FGV - 2009 - MEC - Administrador de Banco de Dados)
A figura a seguir representa parte de um formulário para homepages, gerado por meio do uso de tags HTML.



A sintaxe HTML que gera a figura acima é:

e)
```
<form><p>
<input type="checkbox" NAME="OPCAO" VALUE="op1" checked>Masculino
<input type="checkbox" NAME="OPCAO" VALUE="op2">Feminino</p><p>
<input type="radio" NAME="OPCAO" VALUE="op1">Masculino
<input type="radio" NAME="OPCAO" VALUE="op2" checked>Feminino</p>
</form>
```

d)
```
<form><p>
<input type=“radio" NAME="OPCAO" VALUE="op1">Masculino
<input type=“radio" NAME="OPCAO" VALUE="op2“ checked>Feminino</p><p>
<input type=“select" NAME="OPCAO" VALUE="op1“ checked>Masculino
<input type=“select" NAME="OPCAO" VALUE="op2">Feminino</p>
</form>
```

Resposta: Letra E. Os dois *inputs* de cima são do tipo *checkbox* e o Masculino está com o atributo *checked*. E os dois *inputs* de baixo são do tipo *radio* e o Feminino está com o atributo *checked*.

Resolução de Questões

Provas de TI
Tecnologia da Informação para Concursos

Questão 6 (CESPE - 2010 - TRE-MT - Analista Judiciário - Tecnologia da Informação)
Com relação a HTML (Hyper Text Markup Language) em sua versão corrente 4.01, assinale a opção correta.

a) A marcação <image> é utilizada para incluir imagens em páginas HTML.
b) É possível fazer o navegador web abrir uma nova janela para uma página externa utilizando-se o atributo alt.
c) O trecho abaixo está correto e serve para indicar uma âncora para um endereço de email:
<a="mailto:joao@abc.com.br”>Suporte
d) As marcações HTML <tfoot>, <head> e <tr> são apropriadas para uso em tabelas.
e) O seguinte trecho HTML expressa um formulário que apresenta uma lista selecionável do tipo *drop-down*, com a opção Analista visível.

```
<form action="">
  <p>
    <select name="cargos">
      <option value="gerente">Gerente</option>
      <option value="programador">Programador</option>
      <option selected value="analista">Analista</option>
    </select>
  </p>
</form>
```

Resposta: Letra E. A tag <select> delimita as opções (*tags* option) que juntas formam uma lista de seleção única (*drop-down*).

➢Resolução de Questões

Provas de TI
Tecnologia da Informação para Concursos

Questão 7 (FCC - 2010 - DPE-SP - Agente de Defensoria – Programador)

Em relação ao HTML, considere as colunas abaixo.

I.	Atributo da tag TH para colar, mesclar duas células na vertical.
II.	Campo de formulário que permite selecionar somente uma opção entre várias.
III.	Criar uma nova linha, com cor de fundo verde, em uma tabela (TABLE).
IV.	Equivale a uma imagem com link.
V.	Equivale a uma lista "Ordenada".
VI.	Seleciona uma, nenhuma ou várias opções.

a.	CHECKBOX
b.	<img src="img.gif">
c.	<tr bgcolor='#00FF00'>
d.	RADIO
e.	rowspan='2'
f.	tags OL, LI

A correta associação das colunas é:

a) Ic, IId, IIIf, IVa, Ve, VIb.

b) Ie, IId, IIIc, IVb, Vf, VIa.

c) If, IIa, IIIb, IVe, Vd, VIc.

d) Ie, IIa, IIIf, IVc, Vd, VIb.

e) If, IIa, IIIe, IVc, Vd, VIb.

Resposta: Letra B.

Final da Aula 5

Provas de TI
Tecnologia da Informação para Concursos

GABARITO:

1 – Letra E

2 – Letra B

3 – Letra B

4 – Letra B

5 – Letra E

6 – Letra E

7 – Letra B

Até a próxima aula!