

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA BA - IFBA

APOSTILA: HTML
PROFESSOR: ADDSON ARAÚJO DA COSTA

VALENCA - BA

Sumário

1. Introdução.....	1
2. Tags.....	1
2.1 Tag de abertura e Tag de fechamento.....	1
2.2 Tags únicas.....	1
3. HTML.....	2
3.1 Cabeça, Corpo e Pés.....	2
4. Identação.....	3
5. Tags básicas para manipular texto.....	3
5.1 Parágrafo.....	3
5.2 Cabeçalhos.....	3
5.3 Itálico, Negrito e Sublinhado.....	3
5.4 Linha Horizontal.....	4
5.5 Listas.....	4
6. Atributos.....	4
7. Tabelas.....	4
7.1 Estrutura.....	5
7.2 Atributos (se não sabe o que são atributos, volte ao item 6).....	5
width (Pode ser usado nas tags “table” ou “td”).....	5
height (Pode ser usado nas tags “table” ou “tr”).....	5
border (Pode ser usado na tag “table”).....	5
align (Pode ser usado na tag “td”).....	5
valign (Pode ser usado na tag “tr”).....	5
colspan e rowspan (Pode ser usado na tag “td”).....	6
8. Imagens.....	6
9. Formulários.....	6
9.1 Estrutura.....	6
10. Tipos suportados pela tag input.....	7
10.1 text.....	7
10.2 password.....	7
10.3 checkbox.....	7
10.4 radio.....	7
10.5 date.....	7
10.6 file.....	8
10.7 reset.....	8
10.8 submit.....	8
10.9 image.....	8
10.10 email.....	8
10.11 hidden.....	8
11. Tags de outros campos de entrada.....	8
11.1 select.....	8
11.2 textarea.....	9
12. Fieldset.....	9
13. Iframe.....	9
14. Label.....	9

Lista de Figuras

Figura 1: Uma combobox (esquerda) e um textarea (direita).....8

Figura 2: Exemplo de fieldset.....9

Figura 3: Formulário 2.....10

1. Introdução

Muitos pensam que é difícil criar um website e correm para soluções que entregam o site pronto através de gerenciadores de conteúdo como o Wix, mas os sites ficam todos iguais, sem qualidade gráfica servindo apenas para projetos iniciantes. Outros imaginam ser necessário softwares caros, mas tudo o que você precisa é um browser e um programa editor de textos compatível com programação.

O objetivo dessa apostila é fornecer os conhecimentos básicos partindo do zero. Não é requerido qualquer conhecimento prévio e em breve você verá como é divertido criar websites. Não se aborreça se ficarem feios, pois estamos iniciando nosso curso.

Para criar um website, é necessário o uso da linguagem HTML. Para ver o que estamos falando, acesse uma página WEB qualquer de sua escolha e aperte ctrl + U. Esse código exibido pelo browser é o código HTML do site e o navegador o exibe de forma gráfica de acordo com alguns padrões, por exemplo, o código abaixo serve para exibir a palavra “Exemplo” em negrito.

```
<b> Exemplo </b>
```

Para testar o exemplo acima, crie no seu editor de textos favorito, um arquivo com o conteúdo “ Exemplo ” onde o nome do arquivo é “*Exemplo.html*” e abra esse arquivo no Mozilla Firefox. O resultado deverá ser:

Exemplo

Nessa apostila, todos os códigos com fundo cinza devem ser testados usando essa forma descrita acima e nos pontos onde não há código, está sendo descrito o que deve ser feito, é necessário testar cada tag, cada atributo, cada combinação possível pois só é possível aprender a nossa disciplina com prática. A abordagem teórica dessa apostila é restrita às primeiras páginas.

2. Tags

No HTML, como vimos no exemplo anterior, são usados marcações especiais para definir como é visto o que está escrito. Essas marcações seguem um padrão e são chamadas de “tags”.

Para escrever uma tag de forma correta é preciso escrever onde a Tag inicia e onde ela acaba. No exemplo anterior, a Tag inicia onde está escrito “” e acaba onde está escrito “”. O conteúdo entre a tag de abertura e tag de fechamento é o que será afetado por ela.

2.1 Tag de abertura e Tag de fechamento

O padrão para abrir uma tag é <NomeDaTag> e para fechar é </NomeDaTag>. A diferença entre elas está na presença da barra “/” na tag de fechamento logo após o sinal de menor “<”. Tudo que estiver entre a tag de abertura e fechamento será processado de acordo com a definição da tag.

Note que não está sendo ensinado usando espaços em branco dentro da tag de abertura nem de fechamento, o único caso que é permitido uso de espaço em branco é após o nome da tag, nunca antes.

2.2 Tags únicas

Existe também Tags que abrem e fecham em uma mesma tag, estas seguem o padrão <NomeDaTag/>. Nesse caso, é usado essa outra forma de escrita porque elas não necessitam de conteúdo a processar, são comandos isolados e independentes como, por exemplo, a tag para criar uma quebra de linha cujo código está abaixo. Nesse caso, a quebra de linha funciona independentemente de qualquer texto a processar, ela simplesmente cria uma quebra de linha na posição onde estiver inserida.

```
<br />
```

3. HTML

O HTML é um conjunto de tags, cada uma com sua função. Usando-as juntas podemos obter qualquer resultado visual desejado. O padrão tem uma forma correta de escrita e poderá ter resultados diferentes em cada browser, pois é o browser o responsável por interpretar o html, podendo cada um fazer sua interpretação de forma diferente, contudo existe um padrão aceito pela grande maioria dos browsers que aprenderemos aos poucos.

O padrão não define se os nomes das tags devem ser escritos em minúsculo ou maiúsculo, porém é recomendável utilizar letras minúsculas para maior legibilidade. Use isso como um padrão.

O HTML usa algumas Tags padrão que devem estar presentes só uma vez no seu código. Elas são mostradas abaixo no exemplo. Usamos a tag “html” para identificar onde começa e acaba o documento.

```
<html>
  <head>      </head>
  <body>      </body>
  <foot>      </foot>
</html>
```

Em todos os códigos é necessário o uso das tags acima uma vez que cada uma tem sua função como será mostrado abaixo. Essas tags devem estar na hierarquia mostrada no exemplo acima, pois não faria sentido nenhuma outra combinação. Isto não apenas por significarem cabeça, corpo, pés e sabermos a posição desses itens no corpo humano, mas no HTML isto torna mais rápido e eficiente o processamento para que a página carregue de forma correta seu conteúdo, uma vez que em cada uma dessas tags contem um pedaço específico do documento.

3.1 Cabeça, Corpo e Pés

A tag “head” (cabeça) é usada como se fosse o cérebro do site, isto é, contém informações que não são diretamente visíveis como, por exemplo, elementos de definição como CSS (estética), importação de elementos de frameworks/bibliotecas, título do site, entre outros. Note que nunca devemos inserir tags que geram efeito visual diretamente, mas sim itens que alteram visual de algo ou importam informações externas.

No “body” (corpo) deve conter a parte visível, ou seja, todas as tags que geram efeito visual na tela. Isto abrange quase todas como, por exemplo, a tag “b” que representa o negrito e a tag “br” que efetua a quebra de linha. Veremos mais tags no próximo tópico, todas elas deverão ser colocadas dentro do “body” (corpo), podendo ser combinadas de acordo com seu desejo.

Já na tag “foot” (pé) devemos inserir os códigos que geram movimento/animações (criados em Javascript/Jquery). Em geral, para executar animações, precisamos executar funções que dependem de o documento já estar completamente carregado, por isto, esta tag deve estar por último garantindo assim que todo o conteúdo já foi carregado.

Aqui é comum confundir as tags header e footer com as apresentadas aqui que foram head e foot. Header e footer não são a cabeça e pé, são o cabeçalho e o rodapé e são usadas para propósitos bem diferentes, isto é, topo e rodapé do site e por ter elementos visíveis elas são colocadas dentro do body, mas isto é conteúdo de HTML5 que veremos a frente.

```
<html>
  <head>
    <title> Minha primeira página!! </title>
  </head>
  <body>
```

```
        <b> Funciona!!! \o/ </b>
</body>
<foot>
</foot>
</html>
```

4. Identação

Quando colocamos conteúdo dentro de uma tag no HTML usamos a idéia que a tag passa a ser pai de seu conteúdo e o conteúdo passa a ser filho desta tag.

Esse conceito será muito importante mais a frente. Por enquanto ele será útil para identarmos nosso arquivo corretamente. Identação é o uso do recuo da linha em relação a sua margem.

Deixar esse espaçamento é muito importante pois, nossos códigos, aos poucos, ficarão maiores e precisaremos identificar rapidamente qual elemento é filho de qual. Dessa forma usamos a regra de que todo elemento filho estará um recuo mais a direita que seu pai.

Para realizar esses recuos de forma rápida, ao invés de usarmos a barra de espaço do teclado, devemos usar o botão Tab, localizado a esquerda do teclado ao lado da letra “Q”. Note também que a tag de abertura e fechamento dos pais devem estar com mesma distância de recuo.

Em muitos editores de texto, ao selecionar várias linhas e apertar “tab”, todas as linhas selecionadas irão um espaçamento para a direita e se ao invés de apertarmos “tab” (com as linhas selecionadas), apertarmos “shift” + “tab”, essas linhas perderão um espaçamento.

5. Tags básicas para manipular texto

5.1 Parágrafo

O parágrafo é definido usando a tag “p”. A vantagem de usar a tag de parágrafo ao invés de digitar o texto de outra forma é que após o texto de cada parágrafo, será automaticamente quebrado a linha além de ser possível definir o recuo da primeira linha no parágrafo, mas isso só será possível com CSS, que veremos em seguida. Exemplo:

```
<p> Meu primeiro parágrafo! </p>
```

5.2 Cabeçalhos

No HTML existe seis tags para alterar o tamanho do texto para tamanhos pré-definidos. Estas tags são a “h1”, “h2”, “h3”, ... “h6” e significam, respectivamente, cabeçalho de nível 1, 2, 3 etc, em outras palavras, texto de primeiro nível, segundo nível (menos importante) e terceiro nível (menor ainda) etc.

Essa tag altera o tamanho do texto para configurações fixas com aspecto de negrito. Em geral, estamos interessados em ajustar o tamanho do texto para um valor específico de tamanho. Mais para frente veremos a possibilidade de usar CSS para personalizar essas tags, por enquanto é interessante testar o código abaixo.

```
<h1>Cabeçalho</h1>
<h2>Subtítulo</h2>
<h3>Sub-subtítulo</h3>
```

5.3 Itálico, Negrito e Sublinhado

Assim como podemos usar a tag “b” para transformar um texto em negrito, podemos usar a tag “i” para deixá-lo como itálico ou “u” para sublinhado. Também podemos combiná-los.

```
<i>Este texto deve ser itálico.</i>
<b>Estará em negrito</b>
```

```
<u>E sublinhado</u>
```

5.4 Linha Horizontal

Podemos precisar criar uma linha horizontal para separar conteúdos em um site, a forma mais simples é utilizando a tag “hr”, experimente!

```
<hr />
```

5.5 Listas

Para criar uma lista de marcadores, usamos a tag “ul”, porém não basta, pois após criarmos a lista de marcadores, precisamos informar quais são os itens da lista. Para definir os itens da lista, usamos a tag “li”.

A forma de usar estas tags são conforme o exemplo abaixo. Note que a lista pode ter vários itens de lista (abrindo a tag ul antes de começar a descrever os itens e fechando após todos eles) e cada item de lista tem, entre suas tags de abertura e fechamento, o conteúdo pertencente ao item.

```
<ul>
  <li>Um item de lista</li>
  <li>Outro item de lista</li>
</ul>
```

Existem também as listas numeradas! Experimente no exemplo anterior modificar o uso da tag “ul” (unordered list, do inglês lista não ordenada) por “ol” (ordered list, do inglês lista ordenada).

Lembro que dentro do “li” (item de lista), podemos ter quaisquer elementos o que inclui a possibilidade de colocar uma lista dentro de outra.

6. Atributos

Na tag de abertura podemos também especificar informações que a tag necessita para seu funcionamento. O exemplo mais simples é quando queremos criar um link. Nesse caso a tag que executa esse funcionamento é a tag “a” (a letra “a” é originada da palavra âncora).

Para definirmos um link precisamos informar que queremos criar uma âncora (para isso usamos a tag “a”), precisamos informar qual texto ficará visível para o usuário clicar (colocamos esse texto entre a tag de abertura e fechamento) e precisamos também informar o endereço do site que será aberto quando o usuário clicar (para isso precisamos usar um atributo para a âncora). Veja o exemplo abaixo e depois a descrição.

```
<a href="http://www.google.com"> Clique aqui! </a>
```

Atributos são características associadas a algum elemento. Nesse caso o elemento é a tag “a” do html, o atributo é o “href” que significa o endereço para o qual o link aponta e entre aspas duplas temos o valor deste atributo, neste caso o endereço.

Para definirmos atributos em html usamos a seguinte notação usando a **tag de abertura**:

```
<NomeDaTag atributo1="valor" atributo2="valor" atributo3...>
```

Um outro atributo da tag “a” (âncora), é o target que define onde será aberto o link. Caso o valor de target seja “_blank”, o link abrirá em nova janela. Não confundir “_blank” com “blank”. No primeiro caso o link sempre abrirá em nova janela, no segundo caso abrirá a primeira vez em nova janela e nas demais vezes abrirá na mesma janela que já havia sido aberta previamente, como se estivesse abrindo sempre na janela que você chamou de blank, outras palavras quaisquer podem ser usadas, tendo resultado semelhante.

7. Tabelas

Tabelas são estruturas que precisam, necessariamente que exista um alinhamento

semelhante a um quadriculado, ou seja, a divisão de linhas percorre toda a estrutura criando um alinhamento no seu conteúdo.

7.1 Estrutura

É muito comum e útil uso de tabelas. Em HTML para usarmos tabelas precisaremos usar três tags. A primeira, “table”, define uma tabela. A segunda, “tr”, define uma linha da tabela. A terceira, “td”, define uma coluna da tabela. Para melhor entendimento veja o exemplo abaixo.

```
<table>
  <tr>
    <td> Coluna 1, Linha 1 </td>
    <td> Coluna 2, Linha 1 </td>
  </tr> <tr>
    <td> Coluna 1, Linha 2 </td>
    <td> Coluna 2, Linha 2 </td>
  </tr>
</table>
```

Note que usando essa notação podemos ter qualquer quantidade de linhas e qualquer quantidade de colunas em cada linha, teste o que ocorreria caso o número de colunas seja diferente entre as linhas.

Uma observação importante aqui é que as tags devem estar uma dentro da outra nesta ordem apresentada no exemplo, nunca em ordem diferente. Ou seja, a tag TD precisa estar dentro da tag TR que por sua vez precisa estar dentro de uma tag TABLE. Esta ordem não pode ser invertida de forma nenhuma.

7.2 Atributos (se não sabe o que são atributos, volte ao item 6)

width (Pode ser usado nas tags “table” ou “td”)

O width define a largura da tabela ou da coluna, de acordo com a tag onde está inserido. O valor da largura pode ser um número para que possamos definir a largura em pixels ou um valor seguido do “%” para especificar uma porcentagem de tamanho em relação a largura que o elemento pode ocupar.

height (Pode ser usado nas tags “table” ou “tr”)

Semelhante ao width, porém define a altura.

border (Pode ser usado na tag “table”)

O atributo border é usado diretamente na tag de abertura da tabela. Ele define se a tabela terá borda ou não. Seu valor deve ser um número maior ou igual a zero e indica a espessura da borda.

align (Pode ser usado na tag “td”)

Mais um atributo útil em tabelas é o “align”, ele define o alinhamento do texto (ou imagens) dentro da coluna, isto é, deve ser usado na tag “td”. Seus valores podem ser “left” (alinhar à esquerda), “center” (centralizar) ou “right” para alinhar a direita.

valign (Pode ser usado na tag “tr”)

Semelhante ao align, o valign também efetua alinhamento, porém no conteúdo das linhas e

esse alinhamento é vertical. Seus valores podem ser “top” (topo), “middle” (meio, não confundir com center), bottom (baixo) e baseline (linha base, é a linha onde o texto está alinhado).

colspan e rowspan (Pode ser usado na tag “td”)

Quando se está usando tabelas em algum editor de texto ou de planilhas, temos a opção de unir células, ou seja, fazer com que uma célula ocupe o lugar de duas ou mais. Para fazermos isto em html definimos o número de células que serão unidas horizontalmente através do colspan e o número de células que serão unidas verticalmente através do rowspan. O valor desses atributos é o número de células a ser unidas. Veja o exemplo abaixo.

```
<table>
  <tr>
    <td> coluna 1 </td> <td> coluna 2 </td> <td> coluna 3 </td>
  </tr> <tr>
    <td colspan="2" rowspan="2"> Duas colunas e duas linhas </td> <td> coluna 3</td>
  </tr> <tr>
    <td> coluna 3</td>
  </tr>
</table>
```

8. Imagens

Uma página HTML sem imagens é uma página sem vida. Para colocar imagens usaremos a tag “img”. Essa tag tem a notação abaixo, onde o atributo “src” informa onde está a imagem a ser exibida. Exemplo:

```

```

Caso a imagem esteja na mesma pasta que seu arquivo HTML, basta colocar no “src” o nome da sua imagem (incluindo a extensão, por exemplo, jpg, png, gif, bmp etc). Caso a imagem esteja dentro de uma pasta, coloque primeiro o nome da pasta seguido do caracter barra “/” e depois o nome da imagem. Caso exista mais pastas entre o arquivo HTML e a imagem, coloque o caminho a partir do arquivo HTML.

Pode-se ainda usar o link de uma imagem na internet no src da imagem. Nesse caso o src deve conter o endereço completo incluindo o “http://”, por exemplo:

```

```

9. Formulários

9.1 Estrutura

Ótimo, já sei colocar um bocado de informações no html e ver na minha tela, mas como faço para permitir o usuário do meu site digitar algo para mim? Usando formulários.

Para definir um formulário usaremos a tag “form” e dentro do formulário podemos colocar campos para o usuário digitar, esses campos são criados usando a tag “input”.

Cada “input” corresponde a um campo onde o usuário pode digitar. Existem diversos tipos de input, inicialmente estudaremos os tipos “text” e “submit” que devem ser escolhidos usando o atributo “type”.

O tipo “text” informa ao input que deve-se ali permitir que o usuário digite informações, enquanto o tipo “submit” informa que trata-se de um botão para enviar esses dados para o seu site. Experimente o exemplo de formulário abaixo.

```
<form>
  <input type="text"/>
```

```
<input type="submit"/>
</form>
```

10. Tipos suportados pela tag input

Vimos anteriormente dois tipos de campos “input”, o campo de texto e o botão de submissão. Agora vamos aprender mais alguns tipos de campos input. Para testá-los use a sintaxe normal do input e no atributo “type”, coloque os valores em negrito abaixo. Teste usando diferentes navegadores.

10.1 text

Usado para que o usuário possa inserir qualquer informação textual com tamanho pequeno.

10.2 password

Esse tipo funciona exatamente como o campo de texto, porém o que for digitado é exibido como asteriscos, por segurança.

10.3 checkbox

Esse campo mostra uma caixinha que pode ser marcada ou desmarcada pelo usuário. A idéia é que o usuário pode escolher quais caixinhas marcar, podendo inclusive escolher todas.

10.4 radio

O tipo radio ao invés de mostrar uma caixinha para o usuário marcar, mostrará um círculo a ser marcado. Nesse caso a idéia é exibir várias opções ao usuário que poderá escolher só uma, isto é, só conseguirá marcar um dos círculos. Para que isso seja possível, além de criarmos o campo input com atributo “type” com o valor “radio”, precisamos também usar o atributo “name” para nomear os inputs do tipo radio.

É preciso nomear os inputs pois o usuário pode escolher apenas uma opção dentre um conjunto de opções. Definimos qual o grupo de opções nesse caso colocando elementos com mesmo nome. Veja o exemplo abaixo:

```
<form>
  Sexo:
  <input type="radio" name="sexo"/> H
  <input type="radio" name="sexo"/> M
  <br />
  Solteiro:
  <input type="radio" name="estadoCivil"/> S
  <input type="radio" name="estadoCivil"/> N
</form>
```

10.5 date

Esse input é utilizado para que o usuário possa escolher datas. Seu uso mostra um calendário para ajudar na escolha.

Esse tipo de input ainda não é suportada por todos os navegadores. Isso ocorre porque esta tag faz parte do padrão HTML 5 que é novo. Uma observação importante é que caso o tipo de campo não seja reconhecido pelo browser, ele irá interpretar como sendo um campo de texto. Teste em diferentes navegadores para ver como funciona nos principais.

10.6 file

Usado para que o usuário possa escolher um arquivo de seu computador para enviar.

10.7 reset

O input do tipo reset mostra na tela um botão que quando apertado, apaga os valores do formulário, fazendo os a voltar para como estavam antes do usuário começar a preencher.

10.8 submit

Todo formulário precisa ser enviado para que seus dados possam ser usados. Entraremos em detalhes mais a frente sobre esse processo, mas, para enviar, o usuário precisará clicar em um input do tipo submit (ficará com a aparência de um botão) ou um input do tipo image, mostrado abaixo.

10.9 image

Aqui não estamos falando de como colocar imagens em sites e sim em criar botões com a mesma funcionalidade do tipo “submit”, porém utilizando imagens, isto é, o usuário pode clicar na imagem e terá o mesmo efeito que faria com um botão.

Nesse caso além de criar o atributo “type” com valor “image”, devemos também criar o atributo “src” informando o caminho onde está a imagem assim como faríamos usando a tag “img”.

10.10 email

Esse tipo de input funciona como um campo de texto porém ao clicar em um botão de submit, é verificado se a informação digitada é, de fato, um email.

10.11 hidden

Quando enviamos um formulário, muitas vezes precisamos enviar junto algum campo com informações que o usuário não deve ou não precisa ter acesso, por exemplo, enviar junto ao formulário a quantidade de tempo que o usuário demorou digitando para desconsiderar os dados quando o usuário demorar demais digitando, pois provavelmente será outra pessoa se aproveitando de um computador com permissões que foi esquecido ligado.

11. Tags de outros campos de entrada

Além da tag input, existe outras tags para entrada de dados, veremos elas abaixo.

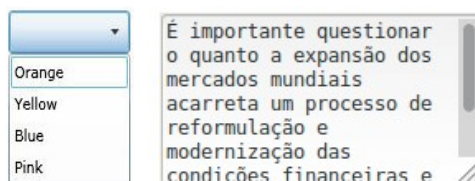


Figura 1: Uma combobox (esquerda) e um textarea (direita).

11.1 select

A tag select cria uma combobox, veja na Figura 1, do lado esquerdo, o que é uma combobox. Para criar uma combobox precisamos definir, também, as opções a ser exibidas quando se clicar na seta ao lado do campo. Para criar um combobox, usamos a tag select e para criar as opções a exibir, usamos a tag option. Veja o exemplo abaixo.

```
<select>
  <option>Volvo</option>
  <option>Saab</option>
  <option>Mercedes</option>
  <option>Audi</option>
</select>
```

11.2 textarea

O textarea cria um campo de texto assim como o input com tipo texto, porém o campo textarea tem tamanho maior para que possa ser digitado textos maiores e o usuário pode redimensionar o textarea facilmente clicando no canto inferior direito. Um exemplo de textarea pode ser visto na Figura 1, do lado direito.

```
<textarea>
  Digite aqui...
</textarea>
```

12. Fieldset

Um site pode ter vários formulários em uma mesma página, é preciso organizá-los para que o usuário saiba quais campos e botões pertencem a cada formulário. Para isso, existe a tag “fieldset” (conjunto de campos) que cria uma borda ao redor dos campos que estiverem dentro dessa tag e usando a tag “legend” podemos definir o nome do formulário que ficará posicionado dentro da borda. É mais fácil entender com um exemplo, execute o código abaixo no seu browser e o resultado deverá ser o mesmo apresentado na Figura 2.

```
<fieldset>
  <legend>Formulário</legend>
  <textarea>
    Digite aqui...
  </textarea>
</fieldset>
```

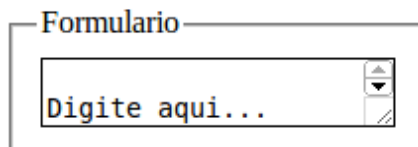


Figura 2: Exemplo de fieldset

13. Iframe

Em alguns casos pode ser útil incluir conteúdo de outro site dentro do seu, para isso usamos a tag iframe. Para usá-la basta usar o atributo src com valor igual ao endereço do site a ser aberto. O iframe funcionará como uma janela para outro site. Além do exemplo abaixo, experimente acessar o youtube e colocar um vídeo dentro do seu site.

Dica: Ao abrir o vídeo no youtube, clique em “share” (compartilhar) e depois em “embed”. Copie o código que o youtube lhe dará e cole em seu site. Ele usará a tag iframe com alguns atributos, tente modificá-los. O que aconteceu?

```
<iframe width="420" height="315" src="//www.youtube.com/embed/4xQb9KI-O3E" frameborder="0"
allowfullscreen>
</iframe>
```

14. Label

Quando se usa botões do tipo radio ou checkboxes torna-se inconveniente para o usuário clicar em um espaço tão pequeno. Como precisamos colocar ao lado desse tipo de botão um texto

indicando o que deve ser preenchido ou o que significa escolher cada opção, não seria ótimo se clicar nesse texto fosse equivalente a clicar no botão? Para isso podemos usar a tag label.

Para que isso funcione é preciso então que a tag label saiba a qual campo ela está relacionada e precisamos também que o campo tenha uma informação única que não pode se repetir no nosso arquivo HTML.

A informação única de cada tag HTML é definida no atributo “id” (um identificador) e na tag label é usado o atributo “for” (do inglês, significa “para”) para indicar, então, o id do campo a que se relaciona.

```
<b> Sexo: </b>
<input type="radio" id="mas" name="sexo"/> <label for="mas">H</label>
<input type="radio" id="fem" name="sexo"/> <label for="fem">M</label>
```

Apesar da tag label ser especialmente útil no caso de inputs pequenos para facilitar o clique, essa tag pode ser usada com qualquer tipo de campo e executará processamento idêntico (equivale a clicar no campo a que ela está relacionada).

Exercício

Com os nossos conhecimentos atuais, conseguimos criar o formulário abaixo. Como você o faria? Monte seu código.

Detalhe 1: Lembre-se de utilizar a tag label sempre que possível para que ao clicar, por exemplo, na palavra “Nome”, seja ativado o input onde se deve digitar o nome.

Detalhe 2: Deve ser possível clicar no botão enviar que deve ser estilizado usando uma figura de botão encontrada na internet, busque uma imagem que lhe seja interessante para um botão.

Detalhe 3: Preste atenção aos alinhamentos. Os campos Input não estão encostando nos sinais de dois pontos exceto no caso do campo “sexo”.

Detalhe 4: Deve ser obedecido a estética idêntica ao exemplo, observe os negritos, itálicos, tamanhos, traços, contorno criado pela tag fieldset, obtenha um resultado idêntico. É isso o que um profissional de design espera de um desenvolvedor.

Formulario de Cadastro:

Nome:	Joaquim	Senha:	
Estado:	RN	Nascimento:	dd/mm/aaaa
Curriculo:	Escolher arquivo Nenhum arquivo selecionado		
☐ H ☐ M			
Não tenho currículo.			
ENVIAR			

Antes de enviar os dados verifique se estão corretos.

- O telefone deve ser fornecido no padrão (xx) xxxx-xxxx
- O CPF deve estar no padrão xxx.xxx.xxx-xx

Figura 3: Formulário 2