

# Aplicações para Internet

## Conteúdo

- HTML5
  - Gráficos e Multimídia
  - Elementos de incorporação
  - Formulários
  - Domínio
  - Hospedagem
  - Cpanel
  - FTP

# Áudio


- Antes do surgimento do HTML5, não existia um padrão para reproduzir som no navegador, o que era possível somente através de plug-ins.
- Na especificação HTML5 existe uma tag que se destina a incorporar um som ou stream (fluxo) de áudio na página sem a necessidade de plug-ins. Essa tag seria `<audio>...</audio>`
- Podemos utilizar um código para todos os formatos suportados de áudio. O comando ou tag para essa tarefa seria: `<source>...</source>` utilizado junto com a tag de áudio.

# Áudio

- Tabela de compatibilidade entre os formatos e os navegadores.

Navegador	MP3	WAV	OGG
Internet Explorer	SIM	NÃO	NÃO
Firefox	SIM	SIM	SIM
Google Chrome	SIM	SIM	SIM
Apple Safari	SIM	SIM	NÃO

Open  
source e  
sem  
patente



- É possível criar os controles com tags HTML e posteriormente programação em JavaScript.

# Áudio

- Tag simples

`<audio src="picapau.mp3" controls="controls"></audio>`

O atributo **controls** adiciona controles como play, pause e volume.

Também: `<audio src="picapau.mp3" type="audio/mpeg" controls />`

Para disponibilizar mais de um formato podemos utilizar a tag source dentro da tag audio:

`<audio controls="controls">`

`<source src="som de derrota.ogg" type="audio/ogg" />`

`<source src="som de derrota.mp3" type="audio/mpeg" />`

`<source src="som de derrota.wav" type="audio/vnd.wave" />`

Seu navegador não aceita a tag audio

`</audio>`



# Vídeo


- Antes do surgimento do HTML5, também não existia um padrão para reproduzir vídeo no navegador, o que era possível somente através de plug-ins.
- Na especificação HTML5 existe uma tag que se destina a incorporar um vídeo ou stream (fluxo) de vídeo na página, sem a necessidade de plug-ins. Essa tag seria `<video>...</video>`
- Também podemos implementar o código para todos os formatos suportados de vídeo. O comando ou tag para essa tarefa seria: `<source>...</source>` utilizado junto com a tag de vídeo.

# Vídeo

- Tabela de compatibilidade entre os formatos e os navegadores

Navegador	MP4	WebM	OGG
Internet Explorer	SIM	NÃO	NÃO
Firefox	SIM	SIM	SIM
Google Chrome	SIM	SIM	SIM
Apple Safari	SIM	NÃO	NÃO

Open  
source e  
sem patente



- Um arquivo de vídeo utiliza compressão e *codecs*, por isso as empresas não concordaram em adotar um único formato.
  - MP4 = codec de vídeo MPEG4 (H.264 para MPEG)
  - WebM = codec de vídeo VP8
  - Ogg = codec de vídeo Theora

# Vídeo

- Tag simples:

```
<video src="video4.webm" controls="controls"  
      type="video/webm" width="400px"></video>
```

ou: `<video src="mov3.mp4" width="400px" type="video/mp4" controls />`

O atributo **controls** adiciona controles como play, pause, volume e solicitar vídeo em tela cheia.

- Para disponibilizar mais de um formato podemos utilizar a tag source dentro da tag *video*:

```
<video poster="video4.jpg" controls width="400px" >  
  <source src="video4.ogv" type="video/ogg" />  
  <source src="video4.webm" type="video/webm" />  
  <source src="video4.mp4" type="video/mp4" />  
  Seu navegador não suporta a tag de video.  
</video>
```



# Incorporando conteúdo

- Já vimos a tag **img**, que incorpora uma imagem na nossa página, porém temos outras tags como:
  - embed (HTML 5)
  - object
- Essas tags incorporam um conteúdo externo às nossas páginas, geralmente estão associados a plug-ins, como Applets, Flash etc.

```
<object width="400" height="50" data="game.swf"></object>
```

```
<object width="100%" height="200px" data="ex1.html"></object>
```

```
<embed width="400" height="50" src="game.swf"/>
```

```
<embed width="100%" height="200px" src="ex1.html"/>
```



# Incorporando janelas

- Podemos incorporar janelas a nossa página para, por exemplo, abrir conteúdos externos. Isso pode ser feito com a tag `iframe`.
- No atributo `src` podemos especificar arquivos html, por exemplo, ou algum URL.

...

Nosso conteúdo <br/><br/>

```
<iframe src="http://www.w3schools.com" width="500px">  
  <p>Seu navegador não suporta iframes.</p>  
</iframe><br/><br/>
```

Mais conteúdo<br/><br/>

```
<iframe src="https://bb.cruzeirodosulvirtual.com.br">  
  <p>Seu navegador não suporta iframes.</p>  
</iframe>
```

...



# Incorporando vídeo externo

- Utilizando a tag `iframe`, podemos incorporar vídeos externos ao nosso site, como por exemplo do YouTube.
- Obs: alguns vídeos possuem direitos autorais. Também conseguimos fazer esse processo com as tags `embed` e `object`, porém o comum seria com `iframe`.

```
<iframe width="854" height="510"  
src="https://www.youtube.com/embed/66f2yP7ehDs"  
frameborder="0" allowfullscreen></iframe>
```



## Tag canvas (2d)

- A tag canvas é utilizada para desenhar imagens dinâmicas através de programação com JavaScript.
- Com a tag canvas e o JavaScript podemos por exemplo, criar gráficos, jogos, animações, etc.
- Praticamente todos os navegadores reconhecem a programação JavaScript utilizando a API (Application Programming Interface ou Interface de Programação de Aplicativos) da tag canvas. As implementações desses códigos são feitas utilizando um contexto de criação gráfica 2d.
- A API da tag canvas possui diversos métodos para a criação de desenhos e manipulação dos elementos inseridos na tag.

# Tag canvas (2d)

```
<!DOCTYPE HTML>
<html>
<head>
<title>Canvas</title>
<style>
canvas{
    background-color:#999999;
    border: 1px solid #F00;
}
</style>
</head>
<body>
<canvas width="400px" height="400px">
    Seu navegador não suporta a tag canvas
</canvas>
</body>
</html>
```

A tag canvas, sem o uso de programação JS, não tem muita utilidade.

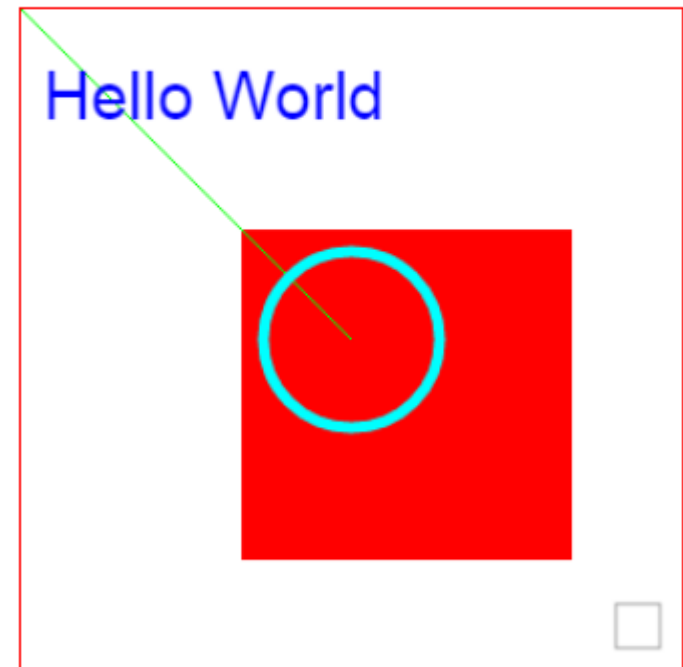


# Tag canvas (2d)

```
<canvas id="meu_canvas" width="300" height="300">  
Seu navegador não suporta a tag canvas  
</canvas>
```

```
<script type="text/javascript">  
  var c=document.getElementById("meu_canvas");  
  var ctx = c.getContext('2d');  
  ctx.fillStyle="#FF0000";  
  ctx.fillRect(100, 100, 150, 150);  
  ctx.strokeStyle="#00FF00";  
  ctx.moveTo(0,0);  
  ctx.lineTo(150,150);  
  ctx.stroke();  
  ctx.font = "30px Arial";  
  ctx.fillStyle="#0000FF";  
  ctx.fillText("Hello World",10,50);  
  ctx.strokeStyle="#AAAAAA";  
  ctx.strokeRect(270,270,20,20);  
  ctx.beginPath();  
  ctx.strokeStyle="#00FFFF";  
  ctx.lineWidth = 5;  
  ctx.arc(150,150,40,0,2*Math.PI);  
  ctx.stroke();  
</script>
```

**Desenhando em um canvas...**



# Tag canvas (2d)

- Para utilizarmos a tag canvas em JavaScript, precisamos dar uma identificação para ela através do atributo ID. O valor desse atributo deve ser único (o nome atribuído a esse atributo não pode se repetir no mesmo arquivo em outros elementos do tipo canvas).
- Alguns métodos para “desenhar” em um canvas são: fillRect, arc, fillText, strokeText, moveTo, lineTo, drawImage etc. Algumas propriedades: fillStyle, font, lineWidth, strokeFill, strokeStyle etc.
- Um bom tutorial em:  
<http://www.w3schools.com/canvas>

```
<!DOCTYPE html><html>
<head>
<style>
  canvas { border: 1px solid #F00; }
</style>
</head>
<body>
  <canvas id="meu_canvas" width="300"
    height="300">
    Seu navegador não suporta a tag canvas
  </canvas>
```

```
<script type="text/javascript">
var c = document.getElementById("meu_canvas");
var contexto = c.getContext('2d');
contexto.fillRect(100, 100, 150, 150);
</script>
</body></html>
```

# Tag canvas (3d) (WebGL)

- A implementação de gráficos 3D utilizando a tag canvas também se dá através de programação JavaScript , porém, neste caso utilizando a API WebGL.
- A API WebGL é derivada da OpenGL ES 2.0, e fornece as mesmas funcionalidades, porém no contexto HTML através da tag canvas.
- Nem todos os navegadores suportam essa API.
- Existem diversas Bibliotecas na Internet,
- No final desta apresentação deixamos alguns links para sites com bibliotecas e exemplos sobre WebGL.

Exemplo: <http://learningwebgl.com/lessons/lesson06/index.html>

# SVG

- SVG é a sigla em inglês para Scalable Vector Graphics ou Gráfico Vetorial Escalável. É uma linguagem de marcação que adota a sintaxe XML e se destina a criar imagens gráficas vetoriais. Esses gráficos não perdem a qualidade se forem redimensionados.
- Podemos criar um arquivo SVG e incluir o mesmo no HTML através das tags
  - embed, object, iframe, img e script.
- Também podemos inserir através da tag `<svg>...</svg>`



# SVG

- Exemplo com o uso da tag `<svg>` dentro do HTML

```
<!DOCTYPE html>
```

```
<html>
```

```
<head><title>SVG</title></head>
```

```
<body>
```

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1" width="400px">
```

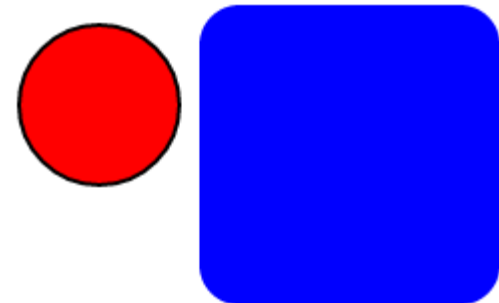
```
<circle cx="100" cy="50" r="40" stroke="black" stroke-width="2" fill="red" />
```

```
<rect x="150" y="0" rx="20" ry="20" width="150" height="150"  
  style="fill:blue;" />
```

```
</svg>
```

```
</body>
```

```
</html>
```



# SVG

- Exemplo com o uso do arquivo externo .svg

Exemplo2.html

```
<!DOCTYPE HTML>
<html>
<head><title>SVG</title></head>
<body>
<object data="exemplo1.svg" type="image/svg+xml" width="500px"></object>
</body>
</html>
```

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <circle cx="125" cy="75" r="60" fill="#FFCC00" />
  <circle cx="225" cy="75" r="60" fill="#3366CC" />
  <circle cx="325" cy="75" r="60" fill="#33FF99" />
</svg>
```

Exemplo1.svg

# Formulários

- Quem nunca preencheu um formulário de cadastro na WEB? A ideia de formulário todos conhecemos. Em resumo, sua principal utilidade é coletar os dados do usuário e fazer algum tipo de processamento no servidor, porém também podemos utilizar para processar algo em JavaScript, no cliente.
- Podemos criar nossos formulários através do HTML utilizando tags para cada tipo de elemento de entrada de dados. No HTML5 também temos alguns atributos que irão facilitar a validação do formulário sem a necessidade de desenvolver scripts em outras linguagens, ou seja, para validar alguns dados não precisaremos utilizar o JavaScript diretamente.

Nome	<input type="text" value="Nome"/>
Email	<input type="text" value="Email"/>
Telefone	<input type="text" value="Telefone"/>
Assunto	<input type="text" value="Assunto"/>
Mensagem	<input type="text" value="Mensagem"/>
	<input type="button" value="Enviar"/>

Email:	<input type="text" value="name@example.com"/>
Password:	<input type="password" value="....."/>
	<a href="#">Esqueceu sua senha?</a>

# Formulários

- Como a linguagem HTML5 ainda está em desenvolvimento, alguns atributos e comandos de formulários não são compatíveis com todos os navegadores, por isso iremos utilizar somente o que for compatível com pelo menos dois navegadores (Google Chrome e Firefox).
- Para criarmos um formulário podemos utilizar a tag <form> ou em alguns casos somente o elemento como um botão para disparar uma ação na nossa página.
- Tag form

<form name="form1" method="post" action="">

Elementos do formulário

</form>

get ou post

Endereço, página  
do servidor

# Formulários (tag input)

- Tag para entrada de dados, que poderá ter diversas variações, conforme exemplos abaixo:
- Exemplos:

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

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

E-mail: `<input type="email" name="usermail" />` (está opção irá validar o campo de e-mail)

URL: `<input type="url" name="homepage" />` (usado para validar URLs)

`<input type="submit" value="Enviar" />` (Para enviar o form para o servidor)

# Formulários (tag input) cont.

- Campo oculto

```
<input name="oculto" type="hidden" value="algumacoisa">
```

- Checkbox (o usuário pode marcar mais de uma opção)

```
<input type="checkbox" name="carros" value="Camaro" />Camaro <br/>  
<input type="checkbox" name="carros" value="Ferrari" />Ferrari
```

- Button para chamar funções em Javascript

```
<input name="botao" type="button" value="Clique aqui" onclick="alert('oi')" />
```

Omitimos a tag label (ver depois) para facilitar o exemplo.

Observe que o valor do argumento *name* é o mesmo para todos os checkboxes.

## Formulários (tag input) cont.

- File para carregar arquivos no formulário  
Escolha um arquivo `<input name="arquivo" type="file" />`
- Image para definir um botão com imagem para enviar o formulário  
`<input name="imagem" type="image" src="images.jpg" />`
- Radio (neste tipo o usuário poderá marcar somente uma opção desde que o name seja o mesmo para os elementos do mesmo grupo)  
`<input type="radio" name="sexo" value="F" />Feminino <br/>`  
`<input type="radio" name="sexo" value="M" />Masculino`

Omitimos a tag label (ver depois) para facilitar o exemplo.

Observe que o valor do argumento *name* é o mesmo para todos os radiobuttons.

# Formulários (tag label)

- Relaciona uma entrada (input) a um rótulo.
- Facilita a usabilidade: ao clicar no rótulo (texto) o campo associado ao mesmo é acionado.
- Pode delimitar o elemento inteiro ou pode ser usado com os atributos for e id, onde o atributo for especifica a qual id o label pertence.

```
<form action="x.asp">  
  <label for="masculino">Masculino</label>  
  <input type="radio" name="sexo" id="masculino" value="m" checked>  
  <br/>  
  <label for="feminino">Feminino</label>  
  <input type="radio" name="sexo" id="feminino" value="f">  
  <br/>  
  <br/>  
  <label for="nome">Nome:  
    <input type="text" id="nome" >  
  </label>  
  <br/>  
  <input type="submit" value="Submit">  
</form>
```



## Comentários sobre os argumentos *id* e *name*

- Nos exemplos anteriores mostramos a utilização dos argumentos ou atributos *id* e *name* nos diferentes tipos de tags *input*.

```
<input type="radio" name="sexo" id="masculino" value="m" checked>
```

```
<input type="text" name="nome" id="nome" />
```

- O atributo **id** é frequentemente utilizado em JavaScript para acessar os elementos do documento (da página) e efetuar algum processamento com os mesmos no computador do cliente.
- O atributo **name** é frequentemente utilizado em elementos de formulários (Form) que serão enviados para um servidor Web, permitindo desta forma o processamento dos dados em alguma “página destino”.
- O atributo **name** possui características de vetor, sendo interessante sua utilização em componentes como listas de itens e grupos de checkboxes, permitindo que a “página destino” consiga recuperar os valores selecionados pelo usuário na “página inicial”.
- Este assunto será aprofundado na disciplina Programação Web.

# Formulário (tag fieldset e legend)

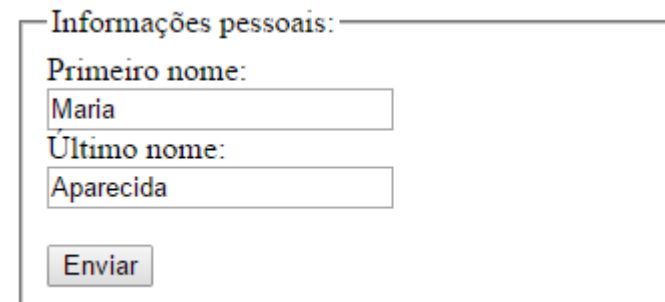
- FIELDSET é utilizado para agrupar elementos ou itens de formulários com características em comum, o Chrome por exemplo, delimita essa tag com uma borda, lembrando que isso pode ser formatado em CSS
- LEGEND é usado para definir uma legenda para o fieldset

...

```
<form action="pagina.php">  
  <fieldset>  
    <legend>Informações pessoais:</legend>  
    Primeiro nome:  
    <br>  
    <input type="text" name="firstname" value="Maria">  
    <br/> Último nome:  
    <br>  
    <input type="text" name="lastname" value="Aparecida">  
    <br/>  
    <br/>  
    <input type="submit" value="Enviar">  
  </fieldset>
```

```
</form>
```

...



# Formulários (tag datalist)

- Define uma lista de opções para um elemento input. Seriam somente sugestões, o usuário poderá digitar valores diferentes.

```
<input list="frutas" name="fruta" />
```

```
<datalist id="frutas">
```

```
<option value="Maçã">
```

```
<option value="Banana">
```

```
<option value="Morango">
```

```
<option value="Abacaxi">
```

```
<option value="Melancia">
```

```
<option value="Pera">
```

```
<option value="Uva">
```

```
</datalist>
```

## Formulários (tag button)

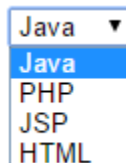
- Nesta tag você poderá colocar texto ou imagens como um botão *clicável*. Esta tag pode ser utilizada fora de um formulário.

```
<button type="button">  
    
</button>
```

# Formulários (tag select)

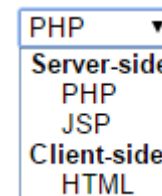
- Cria uma lista de opções (conhecido como combobox)

```
<form>  
  <select>  
    <option value="Java">Java</option>  
    <option value="PHP">PHP</option>  
    <option value="JSP">JSP</option>  
    <option value="HTML">HTML</option>  
  </select>  
</form>
```



Java ▼
Java
PHP
JSP
HTML

```
<form>  
  <select>  
    <optgroup label="Server-side">  
      <option value="PHP">PHP</option>  
      <option value="JSP">JSP</option>  
    </optgroup>  
    <optgroup label="Client-side">  
      <option value="HTML">HTML</option>  
    </optgroup>  
  </select>  
</form>
```



PHP ▼
Server-side
PHP
JSP
Client-side
HTML

## Formulários (tag textarea)



- Define um campo de texto de várias linhas

<form>

<textarea rows="8" cols="70"></textarea>

</form>

# Formulário (alguns atributos comuns)

Atributo	Descrição
disabled	Especifica que o campo está desabilitado, o usuário não pode alterar e o mesmo não é enviado para o servidor
maxlength	Especifica o número máximo de caracteres que podemos digitar no elemento
pattern 	Especifica uma expressão regular, um formato pré-determinado
readonly	Especifica que o campo está somente em modo leitura, semelhante ao disabled, porém neste caso, o valor é enviado para o servidor
required 	Especifica que o campo é obrigatório
size	Especifica o tamanho (largura) do campo em caracteres
value	Especifica um valor padrão inicial para o campo

## Formulários (atributo required)

- Este atributo pode ser utilizado em campos para obrigar a digitação dos dados. A mensagem será ativada quando o usuário tentar enviar o formulário.
- Pode ser utilizado em diversas tags de entrada de dados no formulário.

`<input name="nome" required="required" />`

`<input type="submit"/>`



## Formulários (atributo autofocus)

- Este atributo identifica o elemento ao qual será dado o foco quando a página for carregada. Pode ser utilizado em diversos elementos de um formulário.

```
<form action="x.php">
```

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

```
  Idade: <input type="text" name="idade" /><br />
```

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

```
</form>
```

## Formulários (atributo placeholder)

- Atributo exclusivo de elementos input e textarea, ele define uma dica para digitação no campo, assim que o usuário começar a digitar, a dica é automaticamente retirada.

```
<form action="demo_form.asp">  
  <input type="text" name="user" placeholder="Digite seu Usuário" /><br />  
  <input type="text" name="senha" placeholder="Digite sua Senha" /><br />  
  <input type="submit" value="Submit" />  
</form>
```

# Exemplo desta aula

## Home

[Home](#) | [Blog \(áudio\)](#) | [Página 1 \(vídeo\)](#) | [Página 2 \(íframe\)](#) | [Contato](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce vitae sem eget neque commodo feugiat. Etiam in tempor orci. Suspendisse hendrerit dui ut est facilisis faucibus. Aliquam erat volutpat. Cras facilisis. turpis eget laoreet sodales. lectus massa sollicitudin eros, a scelerisque odio odio vel dui. Donec non erat vehicula, lacina lorem in, tincidunt odio. Nullam in nisl eget mauris sodales aliquam. Donec porta sagittis leo sit amet dapibus. Mauris commodo ipsum at nunc accumsan, sed hendrerit velit molestie.



Legenda da figura

©2017 AI

## Página 2

[Home](#) | [Blog \(áudio\)](#) | [Página 1 \(vídeo\)](#) | [Página 2 \(íframe\)](#) | [Contato](#)

### Video YouTube



### Arquivo html



### Site



©2017 AI

## Blog

[Home](#) | [Blog \(áudio\)](#) | [Página 1 \(vídeo\)](#) | [Página 2 \(íframe\)](#) | [Contato](#)

### Post 1

19 de fevereiro de 2015

Fusce blandit magna at nibh suscipit, vehicula vestibulum justo mollis. In leo mauris, aliquam eu molestie et, venenatis id dui. Duis at tortor nisl. Curabitur bibendum mi et dui blandit interdum. Ut interdum eros id mi rhoncus fermentum. Maecenas ac tortor id eros dapibus rhoncus eget eget eros. Sed lacina portitor erat, eget rutrum urna pulvinar imperdiet. Suspendisse potenti. Morbi vel sem massa. Praesent dictum, augue at pharetra malesuada, mi metus convallis enim, eget pharetra quam nulla ac lectus. Pellentesque tempus convallis libero et varius. Suspendisse viverra tincidunt nulla, consequat tempus est pulvinar ac. Sed sollicitudin portitor viverra.

### Post 2

19 de fevereiro de 2015

Nulla nec lectus sed quam vulputate ornare ac vel orci. Morbi molestie, turpis egestas venenatis efficitur, ipsum leo fringilla neque, id pellentesque purus lorem non felis. Duis id purus diam. Integer efficitur sapien nec mauris dictum elementum. Quisque at nulla semper, rutrum nunc sed, efficitur tortor. Fusce rutrum lectus et dolor tempor tincidunt. Maecenas maximus egestas sodales. Quisque maxims odio a enim posuere fringilla. Aenean a enim nec nisl tincidunt consectetur. Nam eleifend aliquam lorem, et molestie arcu interdum vel. Nullam suscipit placerat velit, at feugiat libero blandit sit amet. Mauris mauris urna, placerat vel orci vitae, condimentum porta ipsum.



©2017 AI

## Página 1

[Home](#) | [Blog \(áudio\)](#) | [Página 1 \(vídeo\)](#) | [Página 2 \(íframe\)](#) | [Contato](#)

### LOREM IPSUM

Donec porta tellus quis nibh dictum, id posuere odio mattis. Sed auctor quis leo sit amet hendrerit. Maecenas molestie enim non nisl tristique tempus. Sed quis ante lorem. Aenean eget tortor metus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae. Morbi tincidunt risus ut ante pellentesque, sed aliquam dolor vulputate. Praesent nibh sem, varius non enim non, faucibus condimentum nisl. Integer a tincidunt augue. Donec nec molestie sem, vel eleifend nisl. Sed ornare vestibulum purus in pretium.



©2017 AI

## Contato

[Home](#) | [Blog \(áudio\)](#) | [Página 1 \(vídeo\)](#) | [Página 2 \(íframe\)](#) | [Contato](#)

### Contato

#### Entre em contato conosco

Preencha o formulário abaixo:

Nome:

Email:

Sexo:  
☒ Masculino  
☐ Feminino

Assunto:

©2017 AI

# Registro de domínio

- Você pode registrar seu Domínio por meio do site Registro.br, após o registro, será cobrada uma taxa anual.
- Outra opção é acessar sites internacionais para fazer o registro ou sites de empresas de hospedagem.



# Hospedagem

- Gratuitos

- Não permite gerenciamento do domínio
- Ferramentas limitadas
- Propagandas
- Sem suporte geralmente
- Servidores instáveis
- Sem backups
- Recursos limitados
- Tráfego limitado

Para sites pessoais ou trabalhos simples

- Pagos

- Gerenciamento completo do domínio
- Diversas ferramentas para gerenciamento
- Suporte
- Servidores estáveis
- Backups dos dados
- Recursos ilimitados
- Tráfego ilimitados

Para sites e serviços profissionais

# Hospedagem

- Gratuitos

<http://www.hostinger.com.br/>



<https://byet.host/>



<https://www.000webhost.com/>

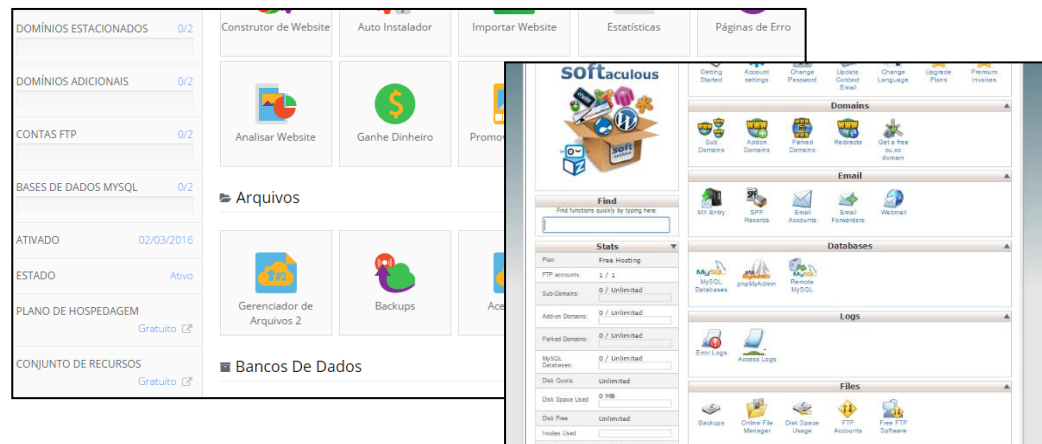


- Pagos



# Cpanel

- Após contratar o serviço, você receberá o URL do seu cpanel (Painel de controle)
  - Exemplo:
    - cpanel.<seudominio>.com
    - cpanel.hostinger.com.br
- Utilize seu usuário e senha para acessar sua área de administração do site
- Procure o File Manager ou Gerenciador de arquivo para “subir” seus arquivos para o servidor
- Insira os arquivos/pastas na pasta **public\_html** ou **htdocs**



# FTP

- Outra forma de colocar os arquivos no servidor seria por FTP/SFTP
- Usando ferramentas de terceiros, geralmente é arrastar e soltar os arquivos/pastas
- Instale a ferramenta FileZilla Client (<https://filezilla-project.org/>)
- Após seu registro, você receberá seu endereço de FTP
  - Exemplo:
    - ftp.<domínio>.com
- Utilize seu usuário e senha para acessar o ftp



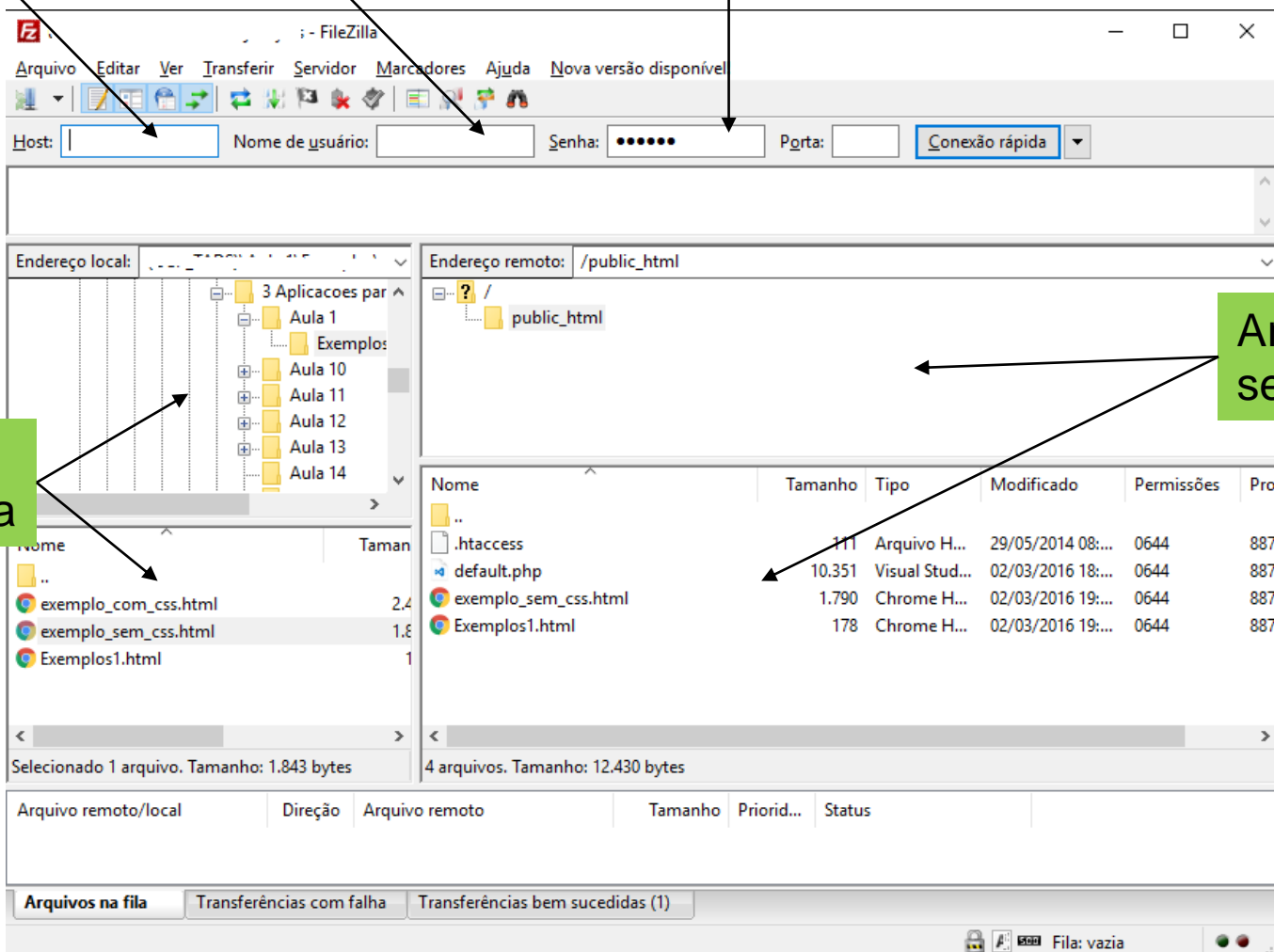


# FileZilla

Seu URL do FTP

Usuário

Senha



Arquivos da  
sua máquina

Arquivos do  
servidor

# Exercícios

1. Crie um pequeno site com 4 páginas, onde:  
Na página index (inicial) insira somente o nome do site (pode ter imagens) e quatro links (HOME | Receita | Vídeo | Contato)
2. Na página Receita insira a descrição completa de uma receita e o componente de áudio para o usuário ouvir a receita, caso a mesma esteja disponível, caso contrário escolha uma música qualquer.
3. Na página Vídeo insira o vídeo relacionado a receita.
4. Na página Contato, crie um formulário aplicando TODOS os elementos estudados nesta aula, inclusive com os atributos passados.

**Aviso: utilize arquivos de áudio e vídeo pequenos, ou seja, até no máximo 5MB cada um. Se preferir insira os vídeos e áudios embutidos na página, assim eles podem ficar disponíveis no YouTube por exemplo.**

# Exercícios

Para incorporarmos os vídeos do YouTube em nossa página, basta procurar o vídeo que queremos, depois clicar com o botão direito do mouse sobre o vídeo e escolher a opção "Ver código de Incorporação", cole o código copiado em seu arquivo html dentro do corpo da página.

Um exemplo de código copiado seria:

```
<iframe width="640" height="360"  
src="http://www.youtube.com/embed/SPFkWkaDLF8?feature=player_detailpag  
e" frameborder="0" allowfullscreen></iframe>
```

**OBS: Como parte do exercício você deverá criar um domínio em algum dos servidores gratuitos comentados na aula e colocar seu exercícios neste servidor, procure colocar dentro de uma pasta como por exemplo, exercícios\_aula3. Após copiar os arquivos no servidor, acesse o seu endereço, tire um print da tela. Compacte esse print com os arquivos do projeto e envie no Blackboard.**

# Links

- <http://www.w3schools.com/html/default.asp>
- <http://www.chromeexperiments.com/webgl/>
- <https://www.khronos.org/registry/webgl/specs/1.0/#1>
- <http://spidergl.org/code.php>
- <http://aleksandarrodic.com/p/jellyfish/>
- <http://www.biodigital.com/biodigital-human.html>
- <http://learningwebgl.com/blog/?p=11>
- <http://www.w3schools.com/svg/>

## Livro sugerido

Mauricio SAMY Silva. **HTML 5 - A Linguagem de marcação que revolucionou a Web**. São Paulo: Novatec, 2011.