

# HTML 5

---

KARINA S. MACHADO

FONTE: W3C SCHOOLS

# Agenda

- Introdução
- Histórico
- Primeiras tags
- Exemplos
- Exercícios

# HTML 5

- HTML 5 será o novo padrão HTML
- A versão anterior do HTML, o HTML 4.01 surgiu em 1999 e já está obsoleta, já que a Internet mudou nesse tempo
- O HTML 5 pretende substituir o HTML 4, o XHTML e o DOM (veremos junto com Javascript)
- HTML 5 tem por objetivo entregar quase tudo que você quer fazer online sem necessitar de plugins adicionais: animações, musica, filmes e outras aplicações complicadas que podem executar no seu navegador.

# HTML 5

- HTML 5 é cross-platform - o que significa que o resultado é mesmo se você está usando um tablet ou smarfone ou um netbook ou uma TV smart
- HTML5 também pode ser utilizado para escrever aplicativos que seguiram funcionando mesmo que você não esteja mais online
- Os grupo de desenvolvedores HTML5 incluem: Apple, Google, IBM, Microsoft, Mozilla, Nokia, Opera, and hundreds of other vendors.
- HTML5 ainda é um trabalho em andamento, entretanto a maioria dos browsers atuais já tem suporte aos elementos HTML5

# Objetivos do HTML5

- Novas características devem ser baseadas no HTML, CSS, DOM e Javascript
- A necessidade por plugins externos como Flash devem ser reduzidos
- A correção de erros deve ser mais fácil que nas versões anteriores
- Scripting deve ser substituído ao máximo por tags html
- O processo de desenvolvimento com HTML deve ser visível ao público

# Vamos iniciar o uso do HTML

- É obrigatório a inclusão da TAG DOCTYPE!
- O Doctype é uma definição do tipo de documento (DTD – Document Type Definition) que possibilita que ferramentas como os navegadores processem de maneira correta esse documento.
- Essa Tag aparece no início da página para indicar ao navegador como processar o documento.
- Definindo o tipo de DOCTYPE você está dizendo qual é a estrutura, quais são os elementos e atributos que esse documento utiliza, isso é necessário pois existem várias versões de HTML, XML, etc...

# Vamos iniciar o uso do HTML

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

# Principais novos elementos/atributos

- O elemento <canvas> para desenhos 2D
- Os elementos <video> e <audio> para execução de mídias na sua página
- Suporte para armazenamento de dados localmente
- Pode funcionar mesmo offline
- Novos elementos de especificar grupos de conteúdos: <article>, <footer>, <header>, <nav>, <section>
- Novos elementos para formulários: calendar, date, time, email, url, search



# Suporte

- Atualmente os elementos de HTML5 são suportados pela maioria dos browsers atuais

## Browser Support



# Elemento CANVAS

- O elemento HTML5 `<canvas>` é utilizado para desenhar gráficos em tempo real (o que geralmente era realizado com scripting JavaScript).
- O elemento `<canvas>` é somente um “espaço” para gráficos. Você utilizará scripts para atualizar o desenho neste “espaço”
- O `<canvas>` tem muitos métodos de desenho: caixas, círculos, texto e adição de figuras.

# Elemento CANVAS

- O elemento `<canvas>` `</canvas>` é uma área retangular para desenho em uma página HTML
- Por default (padrão) esse elemento não tem borda nem conteúdo.
- Principais atributos:
  - Id: identifica o CANVAS → necessário para indicar em que espaço realizar determinado desenho
  - Width: largura
  - Height: altura
- Exemplo de declaração:
  - `<canvas id="myCanvas" width="200" height="100"></canvas>`

# Elemento CANVAS

- Utilize `<style>` para estabelecer características para seu CANVAS:
  - Borda: espessura, cor e tipo de borda.
- Exemplo:
  - `style="border:1px solid #000000;"`
  - De que outra forma posso realizar essa configuração?????

# Exemplo 1

- Vamos criar uma página Web com uma área de CANVAS com as seguintes características:
  - Altura --> 200
  - Largura --> 400
  - Com borda --> `style="border:1px solid #000000;"`
- Observação: não esqueça de iniciar o documento com o `<!DOCTYPE html>`

# Exemplo 1

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<canvas id="meuCanvas" width="400"  
height="200" style="border:1px solid  
#000000;">
```

Seu browser não suporta o elemento canvas.

```
</canvas>
```

```
</body>
```

```
</html>
```

# Desenhando no <canvas> com javascript

Não vamos ver detalhes da linguagem neste momento, só um exemplo de desenho no canvas

Desenhando um retângulo vermelho, crie esse código abaixo do canvas:

- `<script>`
- `var c=document.getElementById("meuCanvas");`
- `var ctx=c.getContext("2d");`
- `ctx.fillStyle="#FF0000";`
- `ctx.fillRect(100,50,200,100);`
- `</script>`

# Explicação

- Para encontrar o elemento canvas:
  - `var c=document.getElementById("meuCanvas");`
- O `getContext("2d")` é um objeto HTML5 built-in com muitas propriedades e métodos para desenhar caixas, círculos, texto, imagens, etc.
  - `var ctx=c.getContext("2d");`
- As próximas linhas desenharam um retângulo e o preenchem de vermelho
  - `ctx.fillStyle="#FF0000";`
  - `ctx.fillRect(100,50,200,100);`
- Voltaremos a esse assunto quando estudarmos javascript

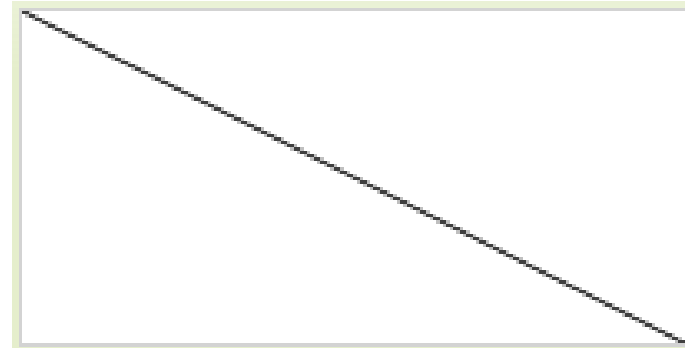


# Resultado

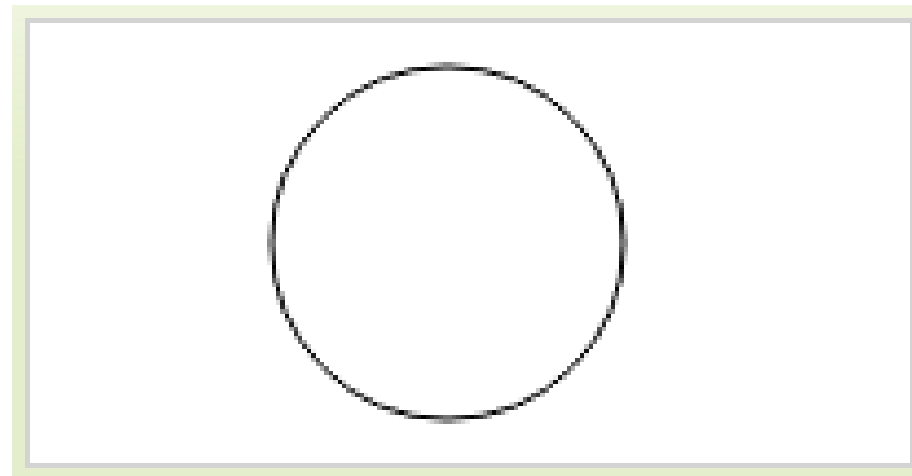


# Alguns outros exemplos:

```
var c=document.getElementById("myCanvas");  
var ctx=c.getContext("2d");  
ctx.moveTo(0,0);  
ctx.lineTo(200,100);  
ctx.stroke();
```



```
var c=document.getElementById("myCanvas");  
var ctx=c.getContext("2d");  
ctx.beginPath();  
ctx.arc(95,50,40,0,2*Math.PI);  
ctx.stroke();
```



# Alguns outros exemplos:

```
var c=document.getElementById("myCanvas");  
var ctx=c.getContext("2d");  
ctx.font="30px Arial";  
ctx.fillText("Hello World",10,50);
```

A rectangular canvas with a light green border. Inside, the text "Hello World" is written in a large, black, sans-serif font, positioned in the upper left area of the canvas.

Hello World

```
var c=document.getElementById("myCanvas");  
var ctx=c.getContext("2d");  
  
// Create gradient  
var grd=ctx.createLinearGradient(0,0,200,0);  
grd.addColorStop(0,"red");  
grd.addColorStop(1,"white");  
  
// Fill with gradient  
ctx.fillStyle=grd;  
ctx.fillRect(10,10,150,80);
```



# Elemento SVG

- SVG padrão para Scalable Vector Graphics
- SVG é utilizado para definir vector-based graphics para a Web.
- SVG define o gráfico utilizando o formato XML
- Gráficos desenhados com o SVG não perdem nenhuma qualidade se aumentados com zoom ou redimensionados
- Todo elemento e todo atributo nos arquivos SVG podem ser animados
- SVG é uma recomendação da W3C

# Elemento SVG - Exemplo

```
<!DOCTYPE html>
<html>
<body>

<svg width="300" height="200">
  <polygon points="100,10 40,180 190,60
10,60 160,180"
  style="fill:lime;stroke:purple;stroke-
width:5;fill-rule:evenodd;" />
</svg>

</body>
</html>
```

# Comparação entre SVG e Canvas

CANVAS	SVG
<p>Dependente da resolução</p> <p>Capacidade ‘pobre’ de renderização de texto</p> <p>Você pode salvar a imagem gerada dentro do canvas como PNG ou JPEG</p> <p>Muito utilizado em jogos</p>	<p>Independente da resolução</p> <p>Boa renderização de texto</p> <p>Fica lenta a renderização se o elemento for muito complexo</p> <p>É estático, não utilizado em jogos</p>

# Elemento Drag and Drop

- “Arrasta e Solta”
  - Tornar um elemento figura “arrastável”:
  - `<img draggable="true">`
- 
- Vamos só executar um exemplo simples:
  - [http://www.w3schools.com/html/html5\\_draganddrop.asp](http://www.w3schools.com/html/html5_draganddrop.asp)

# Video no seu HTML5

- Até a criação dessa Tag HTML5 não havia um padrão para exibir vídeos em um página Web.
- Hoje a maioria dos videos é exibido utilizando um plugin, entretanto, cada browser tem seu plugin e um plugin para cada tipo de video, o que dificultada a publicação de sites com vídeos.
- HTML5 define então um novo elemento que especifica um padrão para exibir videos na página Web, o elemento **<video>**



# Elemento <video>

- Principais atributos:
- Width => largura
- Height => altura
- Controls (para incluir os controles de play, stop, etc)
- Source src => o nome do video a ser exibido (deve estar na mesma pasta do conteúdo da sua página web, da mesma forma que o <img>
- Type => indica o tipo de video (será utilizado pelo browser para corretamente renderizar o vídeo)

# Elemento Video

- ```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
  Your browser does not support the video  
  tag.  
</video>
```
- Vamos ao endereço abaixo para testar (não vamos fazer download de video por isso não vamos testar diretamente)
- [http://www.w3schools.com/html/html5\\_video.asp](http://www.w3schools.com/html/html5_video.asp)

# Elemento <audio>

- Até a criação dessa tag, não havia um padrão para execução de arquivos de audio em páginas Web
- Hoje, assim como para videos, a maioria das páginas se utiliza de plugins para exibir audios em suas páginas => mesmo problema dos videos, diferentes plugins para cada navegador e cada tipo de arquivo audio.
- HTML define um novo elemento que especifica um padrão para embutir um arquivo de audio em uma página Web, o elemento **<audio>**