

# HTML 5



# Apresentação Pessoal

## Vinicius Reis

- Formação Acadêmica
  - Graduação em Tecnologia de Sistemas de Informação – Faculdade São José
  - Pós Graduação em Análise e Projeto de Sistemas – Universidade Gama Filho
- Atuação
  - Analista de Sistemas – SERPRO
- Aprovações
  - SERPRO, DATAPREV, MPU, INPI, Petrobras Distribuidora, entre outras.

# Motivação

	2011	2012	2013	2014	2015
<b>Conceitos Básicos</b>	-	-	1	4	1
<b>Novos Elementos \ Tags</b>	1	4	4	11	-
<b>SVG</b>	-	-	1	-	1
<b>Canvas</b>	-	1	-	2	-
<b>Aplicações Offline</b>	-	-	-	1	1

# Bibliografia Recomendada



**Smashing HTML5: Técnicas para a Nova Geração da Web – Bill Sanders**



**Use a Cabeça! Programação em HTML 5 - Eric Freeman**



**Html 5: a Linguagem de Marcação que Revolucionou a Web - Maurício Samy Silva**

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

[http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)

# Ementa

- Visão Geral
- Histórico de Versões HTML
- O que é o W3C?
- O que é o WHATWG?
- Vantagens de HTML 5
- Tag's Básicas
- Novidades HTML 5
- Canvas
- SVG
- SVG x Canvas

# Ementa

- HTML Multimídia
- Vídeo em HTML 5
- Áudio em HTML 5
- HTML Helpers (plugins)
- Geolocalização
- HTML5 Drag and Drop
- API Storage
- Aplicações Offline

# Visão Geral

## Conceitos Básicos

- **HTML** é a HyperText Markup Language (Linguagem de Marcação de **HiperTexto**).
- Todo conteúdo inserido em um documento web e que tem como característica a interligação a outros documentos web pode ser considerado **hipertexto**.

## Versões

- As 5 versões da linguagem HTML, criada por **Tim Berners-Lee** são: HTML, HTML+, HTML 2.0, HTML 3.0, HTML 3.2, HTML 4.0, HTML 4.01 e HTML 5. As versões HTML e HTML+ são anteriores à criação do W3C.

## Como a linguagem foi criada?

- Tim Berners-Lee usou como base para a criação da HTML a especificação **SGML**, que é um método internacionalmente aceito e que contém normas gerais para a criação de linguagens de marcação.

# Histórico de Versões HTML

Versão	Ano
Tim Berners-Lee cria a WWW	1989
Tim Berners-Lee cria a HTML	1991
Dave Raggett redigiu a HTML+	1993
HTML Working Group define a HTML 2.0	1995
W3C recomenda a HTML 3.2	1997
W3C recomenda a HTML 4.01	1999
W3C recomenda o uso de XHTML 1.0	2000
Primeira Versão Pública da HTML5 WHATWG	2008
HTML5 WHATWG Living Standard	2012
Recomendação Final do W3C para HTML5	2014



# O que é o W3C?

## Definição

- O Consórcio World Wide Web (W3C) é um **consórcio internacional** no qual organizações filiadas, uma equipe em tempo integral e o público trabalham juntos para desenvolver padrões para a Web.

## Missão

- Conduzir a World Wide Web para que atinja todo seu potencial, desenvolvendo protocolos e diretrizes que garantam seu crescimento de longo prazo.



## O Objetivo é a Padronização

- Sites desenvolvidos segundo esses padrões podem ser acessados e visualizados por qualquer pessoa ou tecnologia, independente dos hardware ou software utilizados, como celulares e PDAs, de maneira rápida e compatível com os novos padrões e tecnologias que possam surgir com a evolução da internet.

Os formatos comuns aprovados pela W3C são: **SVG** (desenho vetorial), **PNG**, **HTML**, **XHTML**, **XML** etc.

# O que é o WHATWG?

## Histórico

➤ O **WHATWG** (Web Hypertext Application Technology Working Group) criado em 2004 por desenvolvedores da Apple, da Fundação Mozilla e do navegador Opera, que descontentes com o rumo do W3C, propuseram-se a desenvolver as especificações HTML5, Web Form 2.0 e Web Controls 1.0.

## Atualmente, o que faz o WHATWG?

➤ Em 19 de janeiro de 2011, Ian Hickson, editor da HTML5, publicou no blog da WHATWG uma matéria informando que a especificação para a HTML5 continuaria a ser desenvolvida **exclusivamente** pelo W3C, ficando sob responsabilidade do WHATWG, a continuidade do desenvolvimento de uma especificação para HTML em geral (sem descrição de versão).

# Vantagens de HTML5

## Melhor Funcionamento em Conjunto com CSS e JavaScript

➤ Ao contrário das versões anteriores, o HTML5 fornece ferramentas para a CSS e o Javascript fazerem seu trabalho da melhor maneira possível. O HTML5 permite por meio de suas APIs a manipulação das características destes elementos, de forma que o website ou a aplicação continue leve e funcional.

## Novas Funcionalidades

➤ O HTML5 também cria novas tags e modifica a função de outras. As versões antigas do HTML não continham um padrão universal para a criação de seções comuns e específicas como rodapé, cabeçalho, sidebar, menus e etc. Não havia um padrão de nomenclatura de IDs, classes ou tags. Não havia um método de capturar de maneira automática as informações localizadas nos rodapés dos websites.

# Questão 01

## FCC 2014 – AL-PE – Analista Legislativo – Comunicação Social

O HTML (Hypertext Markup Language) é uma linguagem

- a) que foi usada no início da internet para formatação das páginas, não é mais usada hoje em dia, a não ser em sites construídos há muito tempo.
- b) usada em softwares.
- c) de marcação para sites, blogs e redes sociais que serve para a indexação de conteúdos nestas plataformas e é combinada somente com o PHP (Hypertext Preprocessor).
- d) de marcação que serve de formatação para qualquer página na internet e pode ser combinada, por exemplo, com o CSS (Cascade Style Sheets), PHP (Hypertext Preprocessor), JavaScript, bem como várias outras linguagens informáticas. Atualmente está na versão 5.0 – o HTML 5, usado também em aplicativos para tablets e que tem novas funcionalidades em relação às suas versões anteriores.
- e) não mais usada na internet, que foi substituída por linguagens como Flash, JavaScript e outras, embora continue sendo desenvolvida por programadores, como a versão HTML 5, usada para a programação de aplicativos em tablets.

# Questão 01

## FCC 2014 – AL-PE – Analista Legislativo – Comunicação Social

O HTML (Hypertext Markup Language) é uma linguagem

a) que foi usada no início da internet para formatação das páginas, não é mais usada hoje em dia, a não ser em sites construídos há muito tempo.

b) usada em softwares.

c) de marcação para sites, blogs e redes sociais que serve para a indexação de conteúdos nestas plataformas e é combinada somente com o PHP (Hypertext Preprocessor).

➡ d) de marcação que serve de formatação para qualquer página na internet e pode ser combinada, por exemplo, com o CSS (Cascade Style Sheets), PHP (Hypertext Preprocessor), JavaScript, bem como várias outras linguagens informáticas. Atualmente está na versão 5.0 – o HTML 5, usado também em aplicativos para tablets e que tem novas funcionalidades em relação às suas versões anteriores.

e) não mais usada na internet, que foi substituída por linguagens como Flash, JavaScript e outras, embora continue sendo desenvolvida por programadores, como a versão HTML 5, usada para a programação de aplicativos em tablets.

## Questão 02

**VUNESP 2014 – PRODEST-ES – Tecnologia da Informação - Infraestrutura**

O padrão recomendado pelo W3C para estruturação e apresentação de conteúdo web é o

- a) CSS.
- b) CSV.
- c) XML.
- d) HTTP.
- e) HTML.

## Questão 02

**VUNESP 2014 – PRODEST-ES – Tecnologia da Informação - Infraestrutura**

O padrão recomendado pelo W3C para estruturação e apresentação de conteúdo web é o:

a) CSS.

b) CSV.

c) XML.

d) HTTP.

➡ e) HTML.

## Questão 03

### CESPE 2013 – SERPRO - Analista - Programação Visual

O W3C desenvolve especificações técnicas e orientações por meio de um processo projetado para maximizar o consenso sobre as recomendações, e que garante qualidades técnicas e editoriais e a obtenção do apoio da comunidade de desenvolvedores, do consórcio e do público em geral. O Web Content Accessibility Guidelines (WCAG) e o Web Accessibility Initiative (WAI), desenvolvidos por meio do processo W3C em colaboração com pessoas e organizações em todo o mundo, têm como objetivo a aplicação de definições e de um padrão comum para a acessibilidade de conteúdo web que atenda às necessidades de indivíduos, organizações e governos. Considerando essas informações, julgue os itens subsecutivos.

Web design e aplicações referem-se a padrões para o desenvolvimento de páginas web, como HTML5 CSS, SVG, Ajax e outras tecnologias WebApps.

Certo

Errado



## Questão 03

### CESPE 2013 – SERPRO - Analista - Programação Visual

O W3C desenvolve especificações técnicas e orientações por meio de um processo projetado para maximizar o consenso sobre as recomendações, e que garante qualidades técnicas e editoriais e a obtenção do apoio da comunidade de desenvolvedores, do consórcio e do público em geral. O Web Content Accessibility Guidelines (WCAG) e o Web Accessibility Initiative (WAI), desenvolvidos por meio do processo W3C em colaboração com pessoas e organizações em todo o mundo, têm como objetivo a aplicação de definições e de um padrão comum para a acessibilidade de conteúdo web que atenda às necessidades de indivíduos, organizações e governos. Considerando essas informações, julgue os itens subsecutivos.

Web design e aplicações referem-se a padrões para o desenvolvimento de páginas web, como HTML5 CSS, SVG, Ajax e outras tecnologias WebApps.

 Certo

Errado

# Tag's Básicas

## Estrutura Básica HTML

```
<html>
  <head>
    <title>Curso de HTML5</title>
  </head>
  <body>
    Este é um exemplo da estrutura básica de uma página HTML.
  </body>
</html>
```

- O conteúdo da tag <title>, ou seja, o título da página, será exibido no topo da janela/aba do browser. Quando a página for adicionada aos favoritos, esse título também será sugerido como atalho.
- A tag <body> (corpo) possibilita a definição de propriedades gerais para a página inteira. É possível que a cor do fundo da página ou a cor dos links utilizados seja alterada de forma geral, por exemplo.

# Tag's Básicas

## Relembrando alguns atributos da tag body

- **bgcolor**: define a cor do plano de fundo. Podemos optar pelo código hexadecimal ou pelo nome da cor, assim como em todos os atributos de cores.
- **background**: URL de uma imagem para ser usada como plano de fundo.
- **link**: cor natural dos links, ou seja, enquanto eles ainda não foram clicados (o padrão é azul).
- **alink**: cor dos links quando são clicados (o padrão é vermelho).
- **vlink**: cor dos links após serem visitados (o padrão é roxo).
- **text**: cor do texto da página.

## Exemplificando

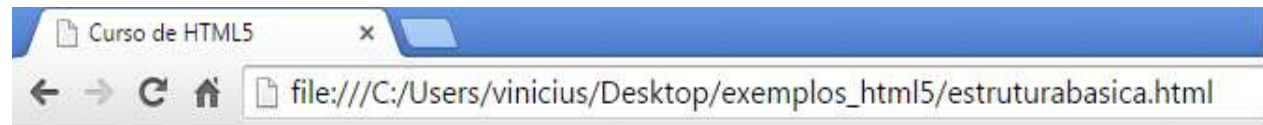
```
<body text="blue" bgcolor="black" link="yellow"></body>
```

# Tag's Básicas

## Cabeçalhos HTML

➤ Em HTML, podemos adicionar cabeçalhos (ou títulos) no corpo da página, diferenciando-os do restante do texto pelo tamanho da fonte. Para isso existem as tags “h”, que possuem seis níveis, de 1 a 6. A seguir tem-se um exemplo de utilização dos seis níveis de cabeçalho:

### Exemplificando



→ **Botafogo**

`<h1>Botafogo</h1>`

**Vasco**

`<h2>Vasco</h2>`

**Fluminense**

`<h3>Fluminense</h3>`

**América**

`<h4>América</h4>`

**Bangu**

`<h5>Bangu</h5>`

**Flamengo**

`<h6>Flamengo</h6>`

# Tag's Básicas

## Modificando a Fonte

➤ Usando a tag <font> é possível alterar algumas das características do texto como o tipo da fonte (Arial, Times New Roman, etc), a cor e o tamanho. Essas propriedades são alteradas com a definição dos atributos face, color e size, respectivamente, da tag font.

## Exemplificando

```
<html>
  <head>
    <title>Curso de HTML5</title>
  </head>
  <body>
    <font face="Arial" size="3" color="blue">Arial 3 Azul</font>
    <font face="Times" size="5" color="green">Times 5 Verde</font>
    <font face="Courier" size="7" color="red">Courier 7 Vermelho</font>
  </body>
</html>
```

# Tag's Básicas

## Resultado do Exemplo Anterior



## Outras Tag's de Formatação

➤ As tag's a seguir servem para apresentar uma formatação adicional ao texto, desde que elas sejam iniciadas antes e fechadas após o texto desejado.

**<b></b>**: marca o texto em negrito (bold).

**<i></i>**: marca o texto em itálico.

**\*<u></u>**: marca um texto como sublinhado.

**\*<s></s> ou <strike></strike>**: risca o texto desejado.

**<sub></sub>**: marca um texto como subscrito.

**<sup></sup>**: marca o texto como sobrescrito.

**\*<center></center>**: centraliza o texto.

**\* Tag's que foram descontinuadas em HTML 5.**

# Tag's Básicas

## Exemplificando as Tag's de Formatação

<b>

negrito

</b>

<i>

itálico

</i>

<sub>

subscrito

</sub>

<sup>

sobrescrito

</sup>

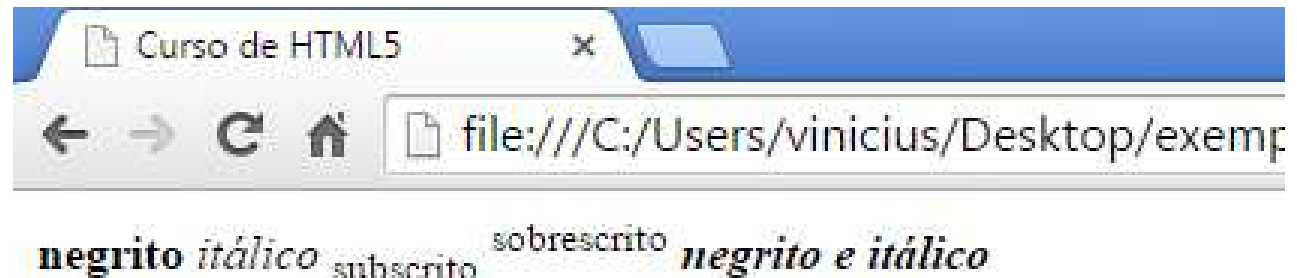
<b>

<i>

negrito e itálico

</i>

</b>



# Tag's Básicas

## Trabalhando com Imagens

➤ Inserir imagens na página é uma das necessidades mais comuns e, por esse motivo, também é consideravelmente simples de ser feito. Para esse fim existe a tag **<img>**, cujos atributos são mostrados a seguir:

**src:** caminho completo do arquivo de imagem (incluindo a extensão do arquivo).

**alt:** texto alternativo para a imagem, geralmente uma descrição da mesma.

**width:** largura da imagem em pixels.

**height:** altura da imagem em pixels.

**IMPORTANTE:** os atributos src e alt são obrigatórios para a tag **<img>**. Além disso, a tag deve ser fechada nela mesma, ou seja, a sintaxe mínima é ****.



# Tag's Básicas

## Exemplificando

```

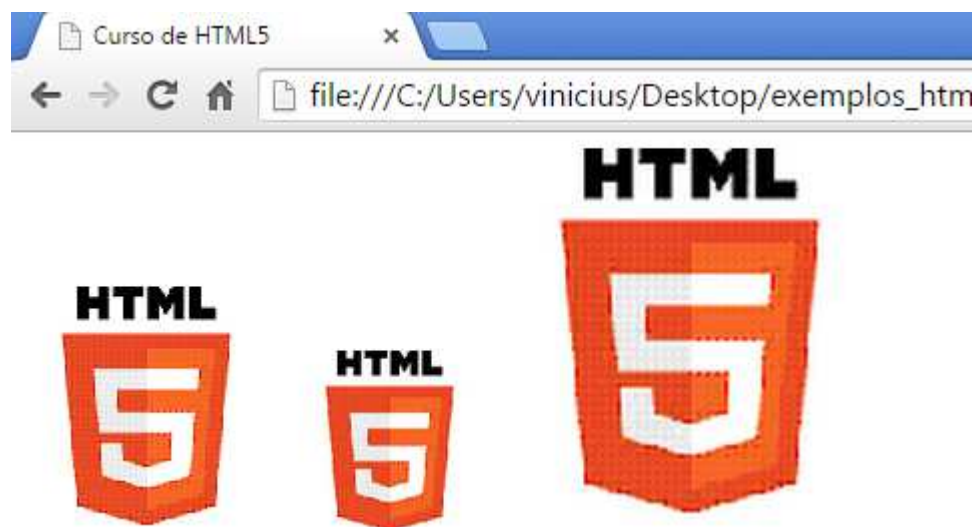
```

```

```

```

```



# Tag's Básicas

## Trabalhando com Links

- As páginas web devem permitir que possamos “navegar” entre conteúdos. Para isso, precisamos descobrir como inserir links para outras páginas, para downloads, e-mails, etc.
- A HTML possui uma tag chamada âncora (anchor, em inglês) representada por `<a>` que nos permite criar links facilmente, apenas definindo seu atributo “href” com o endereço para o qual o usuário deve ser redirecionado.
- A sintaxe básica da tag `<a>` é apresentada a seguir:

**`<a href="https://www.provasdeti.com.br/">Provas de TI</a>`**

- No lugar do texto “Provas de TI”, podem ser inseridos outros elementos como, por exemplo, imagens, títulos, etc. Basta manter a ordem de abertura e fechamento correta das tags.
- No atributo “href” informamos o caminho para o qual o link deve levar o usuário, o que não necessariamente precisa ser outro site.

# Tag's Básicas

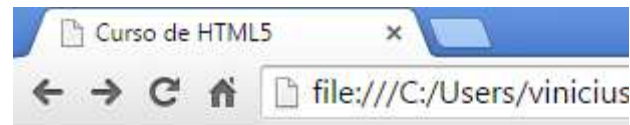
## Exemplificando

```
<a href="http://www.w3c.br/Home/WebHome">W3C</a>
```

```
<a href="arquivo.txt">Download</a>
```

```
<a href="dicas_html5.txt"></a>
```

```
<a href="http://www.w3c.br/Home/WebHome"><h1>W3C</h1></a>
```



[W3C Download](#)

[W3C](#)

# Tag's Básicas

## Trabalhando com Listas

- Outra estrutura bastante comum de ser utilizada em páginas web é a lista. Ela serve pra organizar um conjunto de itens, sequencialmente ou não. As listas podem ser ordenadas ou não, ou seja, cada item podem ou não ter um número/letra/símbolo que o identifique sequencialmente.
- As tags para listas ordenadas e não ordenadas são, respectivamente, `<ol></ol>` e `<ul></ul>`. Entre essas marcações devem ser inseridos os itens, que levam a tag `<li></li>`, conforme os exemplos abaixo:

```
<ol>
  <li>Primeiro item</li>
  <li>Segundo item</li>
  <li>Terceiro item</li>
</ol>
<ul>
  <li>Primeiro item</li>
  <li>Segundo item</li>
  <li>Terceiro item</li>
</ul>
```

# Tag's Básicas

## Resultado do Exemplo Anterior



1. Primeiro item
2. Segundo item
3. Terceiro item

- Primeiro item
- Segundo item
- Terceiro item

## Questão 04

### FGV 2015 – TJ-SC - Analista de Sistemas

Em uma página HTML, deseja-se colocar a figura contida no arquivo "x.png" como link para "www.site.com". O comando HTML a ser usado é:

a `< a href="www.site.com">"x.png"< /a>;`

b `< a href="www.site.com"> < img src="x.png"/>< /a>;`

c `< a link="www.site.com">"x.png"< /a>;`

d `< a link="www.site.com"> < img "x.png"/>< /a>;`

e `< a http="www.site.com">< img file="x.png"/>< /a>.`

## Questão 04

### FGV 2015 – TJ-SC - Analista de Sistemas

Em uma página HTML, deseja-se colocar a figura contida no arquivo "x.png" como link para "www.site.com". O comando HTML a ser usado é:

a `< a href="www.site.com">"x.png"< /a>;`

➡ b `< a href="www.site.com"> < img src="x.png"/>< /a>;`

c `< a link="www.site.com">"x.png"< /a>;`

d `< a link="www.site.com"> < img "x.png"/>< /a>;`

e `< a http="www.site.com">< img file="x.png"/>< /a>.`

## Questão 05

**CESPE 2014 – ANTAQ - Analista Administrativo – Sistemas e Negócios**

Acerca da HTML5, julgue o item subsequente.

No exemplo apresentado a seguir, o atributo hint descreve corretamente a forma de orientar o preenchimento do campo para o usuário.

```
<input type="text" name="Organizacao" hint="ANTAQ">
```

CERTO

ERRADO



## Questão 05

**CESPE 2014 – ANTAQ - Analista Administrativo – Sistemas e Negócios**

Acerca da HTML5, julgue o item subsequente.

No exemplo apresentado a seguir, o atributo hint descreve corretamente a forma de orientar o preenchimento do campo para o usuário.

```
<input type="text" name="Organizacao" hint="ANTAQ">
```

CERTO

➡ ERRADO

## Questão 06

### FCC 2014 – TRT 1ª REGIÃO – Técnico Judiciário - TI

Analise o programa a seguir, elaborado na linguagem HTML.

```
< html>  
  < body>  
    < p>Albert Einstein disse:  
      < q>A imaginação é mais importante que o conhecimento.< /q>  
    < /p>  
  < /body >  
< /html >
```

Quando ele for aberto por um navegador que suporte HTML, será exibido na tela o texto:

- a) Albert Einstein disse: A imaginação é mais importante que o conhecimento.
- b) Albert Einstein disse: **A imaginação é mais importante que o conhecimento.**
- c) Albert Einstein disse: "A imaginação é mais importante que o conhecimento."
- d) Albert Einstein disse: "*A imaginação é mais importante que o conhecimento.*"
- e) "Albert Einstein disse:A imaginação é mais importante que o conhecimento."

## Questão 06

### FCC 2014 – TRT 1ª REGIÃO – Técnico Judiciário - TI

Analise o programa a seguir, elaborado na linguagem HTML.

```
< html>  
  < body>  
    < p>Albert Einstein disse:  
      < q>A imaginação é mais importante que o conhecimento.< /q>  
    < /p>  
  < /body >  
< /html >
```

Quando ele for aberto por um navegador que suporte HTML, será exibido na tela o texto:

- a) Albert Einstein disse: A imaginação é mais importante que o conhecimento.
- b) Albert Einstein disse: **A imaginação é mais importante que o conhecimento.**
- ➡ c) Albert Einstein disse: "A imaginação é mais importante que o conhecimento."
- d) Albert Einstein disse: "*A imaginação é mais importante que o conhecimento.*"
- e) "Albert Einstein disse:A imaginação é mais importante que o conhecimento."

# Novidades HTML 5

## HTML 4.01 x HTML 5

- As principais diferenças entre as duas versões decorrem do fato de HTML 5 ser desenvolvida com o intuito de substituir tanto HTML quanto XHTML, que foi uma tentativa frustrada de reformular o HTML 4 como uma aplicação XML.
- Uma característica importante no desenvolvimento de HTML 5 é a retrocompatibilidade. Na nova versão, os elementos em desuso são considerados apenas obsoletos, mas mesmo assim são suportados.
- Na HTML 5, não existem componentes em desuso (deprecated) com o mesmo sentido que a palavra tem em HTML 4, os elementos são apenas considerados obsoletos (obsoletes).



# Novidades HTML 5

## Doctype

- A estrutura básica do HTML 5 continua sendo a mesma das versões anteriores da linguagem, há apenas uma exceção na escrita do DOCTYPE.
- O DOCTYPE deve ser a primeira linha de código do documento antes da tag HTML.
- O DOCTYPE indica para o navegador e para outros meios qual a especificação de código utilizar. Em versões anteriores, era necessário referenciar o DTD diretamente no código do DOCTYPE. Com o HTML5, a referência por qual DTD utilizar é uma responsabilidade do browser.
- O DOCTYPE não é uma tag do HTML, mas uma instrução para que o browser tenha informações sobre qual versão de código a marcação foi escrita.

# Novidades HTML 5

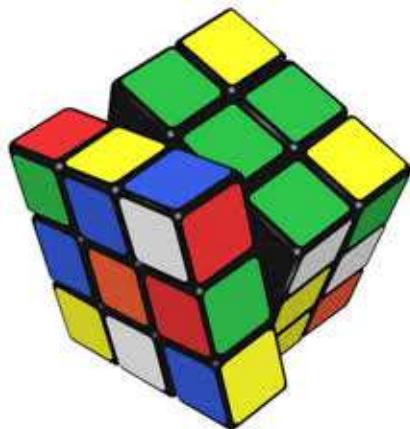
## Exemplificando

- Veja um exemplo de declaração de DOCTYPE em HTML 4.01:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/teste.dtd">
```

- A declaração do DOCTYPE(tipo de documento) em HTML 5 é muito simples:

```
<!DOCTYPE html>
```



simplificando...



E quando o HTML5 mudar, não vamos mudar esse *doctype*, já foi decidido que esse vai ser permanente. Ou seja, uma coisa a menos para se preocupar.

## Questão 07

### **CESGRANRIO 2014 – EPE - Analista de Gestão Corporativa - Tecnologia da Informação**

HTML (HiperText Markup Language) é uma linguagem de marcação inventada por Tim Berners-Lee e se tornou a principal linguagem de criação de páginas Web. Em evolução constante, sua especificação atual é conhecida como HTML 5.

Segundo essa última especificação, um documento HTML, versão 5, básico, deve começar com uma declaração que define seu tipo, que indica ao browser que o conteúdo do elemento deve ser tratado como HTML.

Qual é essa declaração?

a < !DOCTYPE html>

b < ?DOCTYPE html>

c < DOCTYPE html5>

d < ?html version="5.0"?>

e < ?html5 version "1.0"?>

## Questão 07

**CESGRANRIO 2014 – EPE - Analista de Gestão Corporativa - Tecnologia da Informação**

HTML (HiperText Markup Language) é uma linguagem de marcação inventada por Tim Berners-Lee e se tornou a principal linguagem de criação de páginas Web. Em evolução constante, sua especificação atual é conhecida como HTML 5.

Segundo essa última especificação, um documento HTML, versão 5, básico, deve começar com uma declaração que define seu tipo, que indica ao browser que o conteúdo do elemento deve ser tratado como HTML.

Qual é essa declaração?

- ➡ a < !DOCTYPE html>
- b < ?DOCTYPE html>
- c < DOCTYPE html5>
- d < ?html version="5.0"?>
- e < ?html5 version "1.0"?>



# Novidades HTML 5

## Codificação de Caracteres

- Metatag responsável por chavear qual tabela de caracteres a página está utilizando.
- A declaração do CHARSET em HTML 5 também ficou mais simples:

Em **HTML 4.01** era assim:

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

Em **HTML 5** ficou assim:

```
<meta charset="utf-8">
```

**DICA:** Essa forma antiga será também suportada no HTML 5. Contudo, é melhor que você utilize a nova forma

**IMPORTANTE:** A codificação de caracteres padrão do HTML 5 é UTF-8.

# Novidades HTML 5

## Principais API's

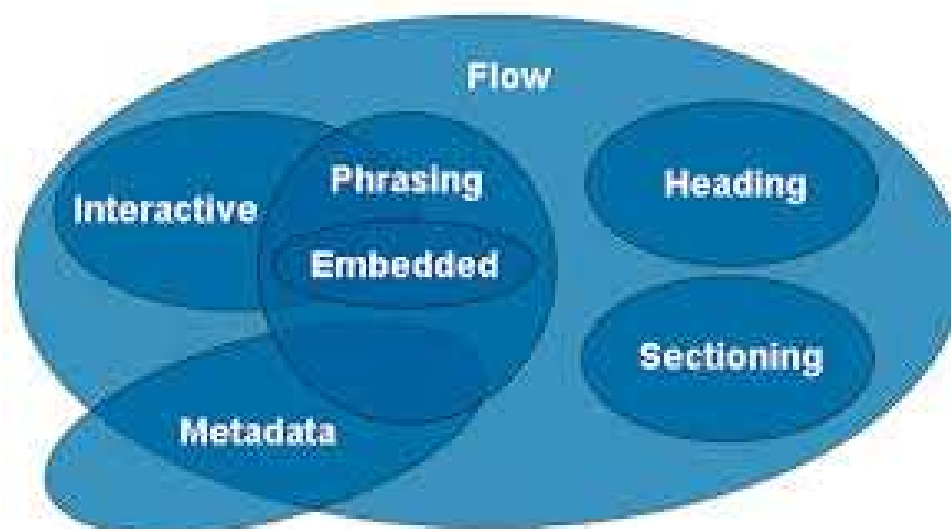
➤ As 6 principais novas API's (Application Programming Interfaces) da HTML 5 são:

- HTML Geolocation
- HTML Drag and Drop
- HTML Local Storage
- HTML Application Cache
- HTML Web Workers
- HTML SSE

# Novidades HTML 5

## Modelo de Conteúdo

- A especificação HTML 5 estabelece um Modelo de Conteúdo que vem a ser a descrição do tipo de conteúdo que se espera ser inserido em cada elemento.
- Cada elemento pertence a uma ou mais categorias que agrupam elementos segundo o seu Modelo de Conteúdo. As categorias são: metadados, fluxo, seção, cabeçalhos, frase, incorporado, interativo e formulário.
- Elementos de formulários, além de poderem pertencer às categorias de **fluxo**, de **frase** e/ou **interativa**, pertencem também à uma categoria específica denominada **formulário**.



# Novidades HTML 5

## Atributos Globais

➤ Atributos globais são aqueles que podem ser usados em todos os elementos da HTML 5. Não só em elementos novos, mas também com os elementos oriundos da HTML 4 e que agora fazem parte da HTML 5.

- **accesskey**: atribui uma tecla de acesso ao elemento. Deve ser informado um caractere ou uma lista de caracteres do teclado, separados por espaço.

## Exemplificando

- `<a href="http://www.w3c.br/Home/WebHome" accesskey="a">W3C</a>`
- `<a href="https://http://www.provasdeti.com.br/" accesskey="b 0">Provas de TI</a>`

**No primeiro exemplo, o site do W3C poderá ser acessado através das teclas ctrl + shift + a e no segundo exemplo, o site do Provas de TI poderá ser acessado através das teclas ctrl + shift + b ou através da tecla 0 (ideal para dispositivos móveis).**

# Novidades HTML 5

▪ **class:** atribui uma classe identificadora para o elemento. É diferente do identificador único id, pois pode ser usado para identificar várias instâncias de um mesmo elemento, bem como identificar elementos diferentes.

## Exemplificando

`<p class="topico1">Texto a ser marcado como tópico 1</p>`

`<h1 class="topico1 subitem2">Cabeçalho de nível 1 e marcado como tópico 1</h1>`

**DICA:** Nos exemplos acima, uma folha de estilo será capaz de identificar os elementos identificados com a classe “topico1” e estilizar todos eles com o mesmo tipo de fonte, por exemplo.



# Novidades HTML 5

▪ **contenteditable**: atributo novo criado em HTML 5 que se destina a permitir que o usuário edite os conteúdos do elemento no navegador. O contenteditable admite os valores true e false. Este atributo foi criado pela microsoft, e suportado pelos seus navegadores por um longo período, e agora está incluído na especificação HTML 5.

**DICA:** O contenteditable é um atributo herdado, ou seja, quando definido com o valor True para um elemento container, faz com que seus elementos descendentes também sejam editáveis.

## Exemplificando

```
<div contenteditable="true">
  <h1>Título 1</h1>
  <p>Parágrafo editável</p>
  <p contenteditable="false">Parágrafo não editável</p>
  
</div>
```

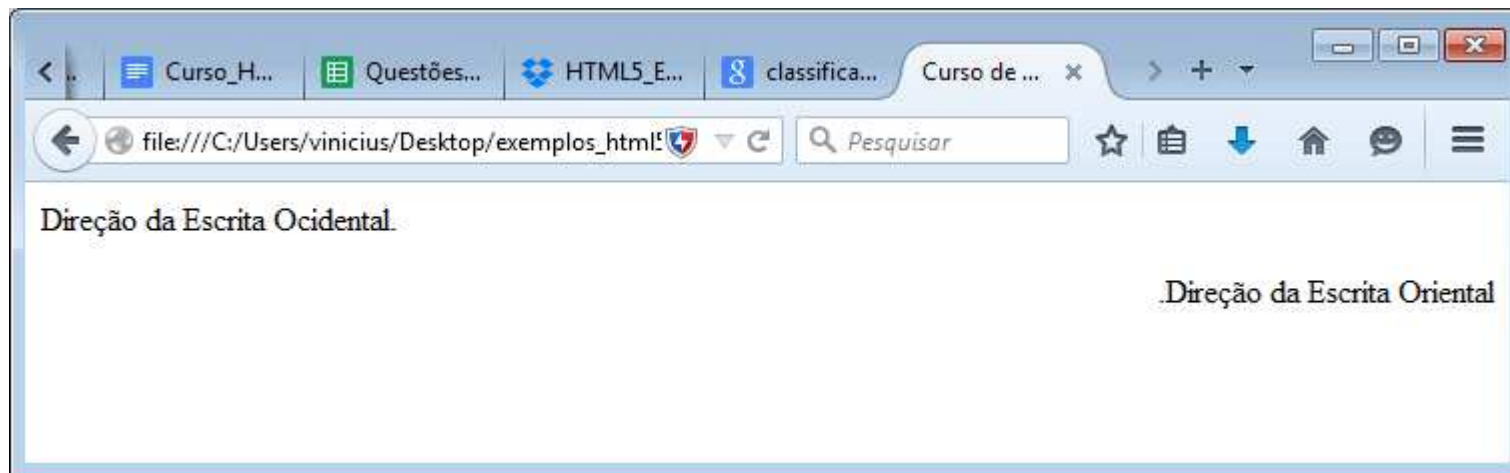
# Novidades HTML 5

- **dir**: atributo que especifica a direção de escrita do texto. Este atributo possibilita dois valores: ltr (left to right) e rtl (right to left).

## Exemplificando

`<p dir="ltr">Direção da Escrita Ocidental.</p>`

`<p dir="rtl">Direção da Escrita Oriental.</p>`



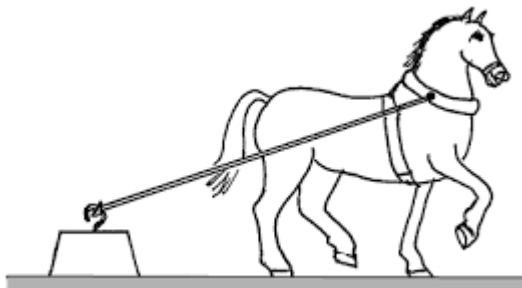
# Novidades HTML 5

▪ **draggable**: atributo novo, criado pela HTML 5 que possibilita que o elemento se torne “arrastável”. Admite os valores “true”, “false” e “auto”. O valor “auto” faz com que o elemento fique com o ajuste padrão do browser utilizado.

## Exemplificando

```
<div id="arraste" draggable="true"></div>
```

**DICA:** No exemplo acima, o elemento div é arrastável na página, mas para isso será necessário estabelecer, com o uso de javascript, as condições de arraste.





# Novidades HTML 5

▪ **hidden**: atributo novo, criado em HTML 5 e que tem a única finalidade de informar se o elemento deverá ou não ser renderizável. É booleano, e não há necessidade de se informar valores, ou seja, se ele estiver presente, o elemento não será renderizado.

## Exemplificando

```
<div id="exemploelementoescondido" hidden>
```

**DICA:** O atributo hidden ainda não é suportado por nenhum navegador atual.

▪ **id**: atributo que possui o objetivo de identificar unicamente um elemento. Cada id deverá ser único por documento. Em HTML 5, a única restrição para a criação de identificadores é a de que os mesmos deverão ter pelo menos um caractere, e quando possuírem mais de um, não poderá existir espaço entre eles.

## Exemplificando

```
<p id="exemplo-de-identificador-valido">
```

# Novidades HTML 5

- **item**: é um atributo destinado aos microdados (extensão de microformats) e se destina a fornecer um mecanismo complementar de adição semântica a documentos HTML, destinada à leitura por máquinas, tais como robôs de indexação.

- **lang**: é um atributo que tem o objetivo de definir o idioma do conteúdo de um elemento. Se este atributo não estiver definido, o idioma assumido será o mesmo do seu elemento-pai.

**DICA:** É uma boa prática declarar o atributo **lang** no elemento-raiz **html**.

## Exemplificando

```
<html lang="pt-br">
```

- **spellcheck**: mais um atributo novo, exclusivo da HTML 5 e que ainda não é suportado por browsers. Seu objetivo é disponibilizar funcionalidades de correção ortográfica e gramatical para os conteúdos textuais inseridos.

# Novidades HTML 5

## Exemplificando

```
<textarea rows="8" cols="40" spellcheck="true"></textarea>
```

- **style**: define regras de estilo para o elemento em que é aplicado, tais como: cores, tamanho de letras, fontes, posicionamentos, entre outras características visuais.

## Exemplificando

```
<p style="color:blue;">Exemplo de parágrafo azul</p>
```

- **tabindex**: tem o objetivo de definir a ordem de tabulação, ou seja, a sequência em que os elementos receberão o foco quando o usuário navegar na página.

## Exemplificando

```
<p tabindex="3">Aqui temos um parágrafo</p>
```

```
<a href="#" tabindex="2">link</a>
```

```

```

# Novidades HTML 5

## Elementos Removidos

Elemento	Trocar Por
<acronym>	<abbr>
<applet>	<embed> ou <object>
<basefont>	CSS
<big>	CSS
<center>	CSS
<dir>	<ul>
<font>	CSS
<frame>	
<frameset>	
<noframes>	
<strike>	CSS
<tt>	CSS

## Questão 08

**FCC 2014 – SABESP - Analista de Gestão - Sistemas**

São elementos da linguagem HTML 4.01 que não fazem mais parte da HTML5, ou seja, não são suportados em HTML5:

a applet e iframe.

b font e frame.

c center e option.

d acronym e button.

e frameset e fieldset.

## Questão 08

**FCC 2014 – SABESP - Analista de Gestão - Sistemas**

São elementos da linguagem HTML 4.01 que não fazem mais parte da HTML5, ou seja, não são suportados em HTML5:

a applet e iframe.

➡ b font e frame.

c center e option.

d acronym e button.

e frameset e fieldset.

## Questão 09

### FEPESE 2014 - MPE-SC - Analista - Tecnologia da Informação

Identifique quais dos seguintes elementos HTML não são suportados pelo HTML5 (possuem substitutos que realizam a mesma função ou função similar no HTML5).

1. < caption >
2. < center >
3. < font >
4. < frame >
5. < img >

Assinale a alternativa que indica todos os itens corretos.

- a São corretos apenas os itens 1, 2 e 3.
- b São corretos apenas os itens 1, 3 e 5.
- c São corretos apenas os itens 2, 3 e 4.
- d São corretos apenas os itens 2, 3 e 5.
- e São corretos apenas os itens 3, 4 e 5.


## Questão 09

### FEPESE 2014 - MPE-SC - Analista - Tecnologia da Informação

Identifique quais dos seguintes elementos HTML não são suportados pelo HTML5 (possuem substitutos que realizam a mesma função ou função similar no HTML5).

1. < caption >
2. < center >
3. < font >
4. < frame >
5. < img >

Assinale a alternativa que indica todos os itens corretos.

- a São corretos apenas os itens 1, 2 e 3.
- b São corretos apenas os itens 1, 3 e 5.
-  c São corretos apenas os itens 2, 3 e 4.
- d São corretos apenas os itens 2, 3 e 5.
- e São corretos apenas os itens 3, 4 e 5.



## Questão 10

### FCC 2014 – TRF 4ª Região – Tecnologia da Informação

A especificação da W3C para a versão 5 da linguagem HTML NÃO traz um conjunto de elementos que eram utilizados na versão anterior, e que são considerados obsoletos, como, por exemplo, os elementos

- a) < frame>, < frameset >, < noframes > e < font >.
- b) < table >, < th >, < tr > e < td>.
- c) < meta >, < big >, < link > e < center >.
- d) < acronym >, < applet >, < b> e < div >.
- e) < frame >, < frameset >, < iframe> e < i >.

## Questão 10

### FCC 2014 – TRF 4ª Região – Tecnologia da Informação

A especificação da W3C para a versão 5 da linguagem HTML NÃO traz um conjunto de elementos que eram utilizados na versão anterior, e que são considerados obsoletos, como, por exemplo, os elementos

➡ a) < frame>, < frameset >, < noframes > e < font >.

b) < table >, < th >, < tr > e < td>.

c) < meta >, < big >, < link > e < center >.

d) < acronym >, < applet >, < b> e < div >.

e) < frame >, < frameset >, < iframe> e < i >.

# Questão 11

**CESPE 2014 – ANTAQ - Analista Administrativo - Sistemas e Negócios**

Acerca da HTML5, julgue o item subsequente.

Alguns elementos do HTML5, tais como o center e o font, são considerados obsoletos por serem utilizados meramente para formatação, função que pode ser realizada por meio de CSS.

Certo

Errado

# Questão 11

**CESPE 2014 – ANTAQ - Analista Administrativo - Sistemas e Negócios**

Acerca da HTML5, julgue o item subsequente.

Alguns elementos do HTML5, tais como o center e o font, são considerados obsoletos por serem utilizados meramente para formatação, função que pode ser realizada por meio de CSS.

➡ Certo

Errado

# Novidades HTML 5

## Elementos de Estrutura – Parte 1 de 3

TAG	Descrição
<article>	Define um artigo no documento
<aside>	Define um conteúdo além do conteúdo da página
<bdi>	Define uma parte do texto que pode ser formatada em uma direção diferente de outro texto
<details>	Define detalhes adicionais que o usuário pode visualizar ou esconder
<dialog>	Define uma caixa de diálogo ou janela
<figcaption>	Define uma legenda para um elemento <figure>
<figure>	Define conteúdo como ilustrações, diagramas, fotos, etc.
<footer>	Define o rodapé para um documento ou uma seção

# Novidades HTML 5

## Elementos de Estrutura – Parte 2 de 3

TAG	Descrição
<header>	Define o cabeçalho para um documento ou uma seção
<main>	Define o conteúdo principal do documento
<mark>	Define marcações ou texto destacado
<menuitem>	Define um item de comando ou menu que o usuário pode invocar a partir de um popup
<meter>	Define uma medida escalar dentro de um intervalo conhecido
<nav>	Define links de navegação no documento
<progress>	Define o progresso da tarefa
<rp>	Define o que mostrar em browsers que não suportam anotações ruby

# Novidades HTML 5

## Elementos de Estrutura – Parte 3 de 3

TAG	Descrição
<rt>	Define uma explicação dos caracteres para a tipografia da Ásia Oriental
<ruby>	Define uma anotação Ruby para a tipografia da Ásia Oriental
<section>	Define uma seção no documento
<summary>	Define uma posição visível para um elemento <details>
<time>	Define a data/hora
<wbr>	Define uma possível quebra de linha

# Novidades HTML 5

## Elementos de Formulário

TAG	Descrição
<datalist>	Define opções pré-definidas para controles input
<keygen>	Define um campo gerador de par de chaves (para formulários)
<output>	Define o resultado de um cálculo



# Novidades HTML 5

## Input Types

INPUT TYPES	Atributos
<ul style="list-style-type: none"><li>•color</li><li>•date</li><li>•datetime</li><li>•datetime-local</li><li>•email</li><li>•month</li><li>•number</li><li>•range</li><li>•search</li><li>•tel</li><li>•time</li><li>•url</li><li>•Week</li></ul>	<ul style="list-style-type: none"><li>•autocomplete</li><li>•autofocus</li><li>•form</li><li>•formaction</li><li>•formenctype</li><li>•formmethod</li><li>•formnovalidate</li><li>•formtarget</li><li>•height and width</li><li>•list</li><li>•min and max</li><li>•multiple</li><li>•pattern (regexp)</li><li>•placeholder</li><li>•required</li><li>•step</li></ul>

# Novidades HTML 5

## Nova Sintaxe de Atributos

➤ HTML 5 permite 4 diferentes sintaxes para atributos. Este exemplo demonstra as diferentes sintaxes usadas na tag `<input>`.

TYPE	Exemplo
Empty	<code>&lt;input type="text" value="João da Silva" disabled&gt;</code>
Unquoted	<code>&lt;input type="text" value=João&gt;</code>
Double-quoted	<code>&lt;input type="text" value="João da Silva"&gt;</code>
Single-quoted	<code>&lt;input type="text" value='João da Silva'&gt;</code>

# Novidades HTML 5

## Gráficos

TAG	Descrição
<canvas>	Define uma forma (desenho) usando JavaScript
<svg>	Define uma forma (desenho) usando SVG

## Mídia

TAG	Descrição
<audio>	Define um conteúdo sonoro
<embed>	Define containers para aplicações externas (como plugins)
<source>	Define códigos para <video> e <audio>
<track>	Define faixas para <video> a <audio>
<video>	Define conteúdo de vídeo

# Novidades HTML 5

## Novos Elementos

*Após o estudo dos atributos globais, vamos passar pela lista de novos elementos da HTML 5.*

- **article**: o elemento `<article>` especifica conteúdo independente e autosuficiente. Um artigo deve fazer sentido por si só e deve permitir assim, a sua reutilização. Um elemento `<article>` pode ser usado em fóruns de discussão pública, posts de blogs ou via RSS.

## Exemplificando

`<article>`

`<article> //Comentário 1 sobre a matéria </article>`

`<article> //Comentário 2 sobre a matéria </article>`

`</article>`

# Novidades HTML 5

▪ **aside:** O elemento `<aside>` define algum conteúdo separado, mas tangencialmente relacionado com o conteúdo ao redor dele. Funciona como os blocos de conteúdos nas barras laterais. É recomendável a utilização de textos adicionais, mas que, se retirados, não causem prejuízos ao texto original.

## Exemplificando

```
<p>Eu e minha família visitamos o Epcot Center neste fim de semana.</p>
```

```
<aside>
```

```
<h4>Epcot Center</h4>
```

```
<p>O Epcot Center é um parque temático localizado na Disney World (Florida).</p>
```

```
</aside>
```



## Questão 12

### FCC 2012 - TRF - 2ª REGIÃO - Técnico Judiciário - Informática

Pedro está criando a página de um artigo de revista utilizando HTML5 e deseja colocar um pequeno texto relacionado ao assunto principal do artigo à esquerda, circundado pelo texto principal. Consultando a documentação oficial da HTML5, Pedro decidiu usar o elemento cuja descrição é apresentada a seguir: ,

This element represents a section of a page that consists of content that is tangentially related to the content around the element, and which could be considered separate from that content. Such sections are often represented as sidebars in printed typography.

Com base na descrição é possível concluir que o elemento escolhido por Pedro foi:

- a <acronym>
- b <article>
- c <sidebar>
- d <aside>
- e <section>


## Questão 12

### FCC 2012 - TRF - 2ª REGIÃO - Técnico Judiciário - Informática

Pedro está criando a página de um artigo de revista utilizando HTML5 e deseja colocar um pequeno texto relacionado ao assunto principal do artigo à esquerda, circundado pelo texto principal. Consultando a documentação oficial da HTML5, Pedro decidiu usar o elemento cuja descrição é apresentada a seguir: ,

This element represents a section of a page that consists of content that is tangentially related to the content around the element, and which could be considered separate from that content. Such sections are often represented as sidebars in printed typography.

Com base na descrição é possível concluir que o elemento escolhido por Pedro foi:

- a <acronym>
- b <article>
- c <sidebar>
-  d <aside>
- e <section>

## Questão 13

### **CESPE 2012 - TJ-RO - Analista Judiciário - Análise de Sistemas - Desenvolvimento**

No que concerne aos elementos da HTML5 e suas funções, assinale a opção correta.

- a O elemento wbr permite representar um grupo de introdução ou elementos de navegação, o que possibilita agrupar índices de conteúdos ou campos de formulários.
- b O elemento datalist representa uma parte da página que pode ser distribuída e reutilizada em serviços de RSS.
- c O elemento aside é um bloco de conteúdo que pode ser representado em textos impressos, em anúncios de publicidade ou mesmo na criação de um grupo de elementos nav e outras informações separadas do conteúdo principal do website.
- d O elemento summary define uma nova seção genérica no documento, possibilitando a divisão da página em várias seções, como introdução, destaque, novidades, informação de contato ou chamadas para conteúdo interno.
- e O elemento track consiste em um grupo de títulos, que permite agrupar elementos de título de H1 a H6 em múltiplos níveis, como título e subtítulos.



## Questão 13

### **CESPE 2012 - TJ-RO - Analista Judiciário - Análise de Sistemas - Desenvolvimento**

No que concerne aos elementos da HTML5 e suas funções, assinale a opção correta.

a O elemento wbr permite representar um grupo de introdução ou elementos de navegação, o que possibilita agrupar índices de conteúdos ou campos de formulários.

b O elemento datalist representa uma parte da página que pode ser distribuída e reutilizada em serviços de RSS.

➡ c O elemento aside é um bloco de conteúdo que pode ser representado em textos impressos, em anúncios de publicidade ou mesmo na criação de um grupo de elementos nav e outras informações separadas do conteúdo principal do website.

d O elemento summary define uma nova seção genérica no documento, possibilitando a divisão da página em várias seções, como introdução, destaque, novidades, informação de contato ou chamadas para conteúdo interno.

e O elemento track consiste em um grupo de títulos, que permite agrupar elementos de título de H1 a H6 em múltiplos níveis, como título e subtítulos.

# Novidades HTML 5

- **command**: tem por objetivo a marcação de um comando evocado pelo usuário. Um comando pode fazer parte de um menu de contexto ou barra de ferramentas criada com o elemento menu ou ainda ser um comando isolado na página. Este elemento deverá ser renderizado como elemento-filho do elemento menu. Atualmente, nenhum navegador possui suporte para este elemento.

O elemento command admite os seguintes atributos: atributos globais, type, label, icon, disabled, checked e title.

- **menu**: seu objetivo é marcar uma lista de comandos, sendo usado como um container para elementos command. Assim como o elemento command, não possui suporte de nenhum browser.

## Exemplificando

```
<menu type="toolbar">  
  <command type="radio" radiogroup="alinhamento" label="esquerda">  
  <command type="radio" radiogroup="alinhamento" label="direita">  
</menu>
```

# Novidades HTML 5

▪ **datalist**: elemento criado para ser usado em conjunto com o elemento *input*. Funciona como uma lista de sugestões para preenchimento do elemento *input* e auxilia o usuário no momento da entrada dos dados.

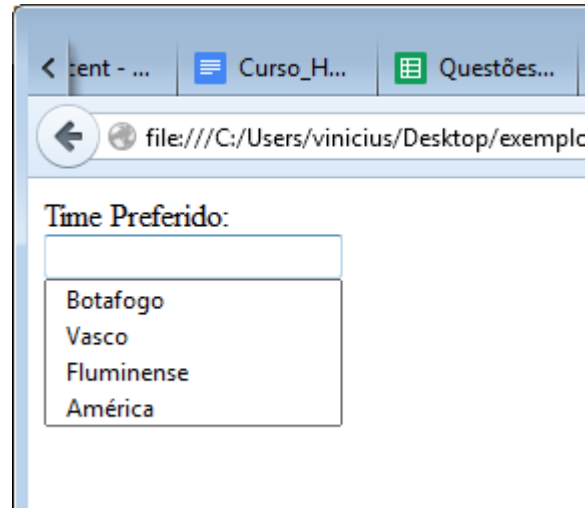
**DICA:** Os navegadores que não oferecem suporte ao elemento `datalist`, renderizam normalmente o elemento `input`, por se tratar de um elemento previsto em edições anteriores da HTML.

## Exemplificando

```
<label for="time">Time Preferido:</label><br>
<input type="text" list="times" id="time">
  <datalist id="times">
    <option value="Botafogo">
    <option value="Vasco">
    <option value="Fluminense">
    <option value="América">
  </datalist>
```

# Novidades HTML 5

## Resultado do Exemplo Anterior



- **details:** destina-se a fornecer informações textuais ou controles de formulário adicionais sobre determinada página ou sobre um conteúdo específico da página. Por padrão, os conteúdos fornecidos por esse elemento não são colocados à vista de um usuário, e o acesso a eles se dá mediante ação do usuário, normalmente por clique em um elemento.

# Novidades HTML 5

▪ **summary**: foi criado para uso em conjunto com o elemento *details* e destina-se a fornecer um rótulo ou um breve sumário das informações adicionais fornecidas por *details*. É um primeiro “filho” do elemento *details* que admite apenas atributos globais.

## Exemplificando

**<details>**

**<summary>**Explicando details**</summary>**

**<d1>**

**<dt>**Explicação assunto1**</dt>**

**<dd>**Explicação assunto1 subassunto1**</dd>**

**<dt>**Explicação assunto2**</dt>**

**<dd>**Explicação assunto2 subassunto1**</dd>**

**</d1>**

**</details>**

**OBS:** Os browsers atuais ainda não oferecem suporte para as tag's *details* e *summary*.

# Novidades HTML 5

- **figure**: esse elemento serve de container para conteúdos independentes relacionados a um conteúdo específico no fluxo do documento. Ele serve para marcar imagens, ilustrações, diagramas, fotos, gráficos, vídeos, entre outros. O elemento figure admite apenas atributos globais e poderá ser usado em conjunto com o elemento *figcaption*.
- **figcaption**: marca uma legenda para o documento inserido com uso do elemento figure. Assim como *figure*, admite apenas atributos globais.

## Exemplificando

```
<figure>  
    
  <figcaption>  
    HTML 5 não é mais problema.  
  </figcaption>  
</figure>
```

# Novidades HTML 5

▪ **footer:** serve para a marcação do rodapé de uma seção ou página. Deve conter informações relevantes, tais como: autor, direitos autorais e similares. Admite apenas o uso de atributos globais.

Em HTML 5, uma página poderá conter mais de um rodapé, pois é válido que cada seção da página tenha seu próprio rodapé.

**CURIOSIDADE:** Nada impede que o rodapé seja colocado no início da seção, embora o recomendado seja o seu posicionamento no fim da seção.

## Exemplificando

```
<article>
  <!-- insira aqui o artigo -->
  <footer>
    <!-- rodapé do artigo -->
  </footer>
</article>
```

# Novidades HTML 5

▪ **header:** destina-se a marcar o cabeçalho de uma seção ou página como um todo e é recomendado que contenha os elementos h1-h6 e informações sobre o conteúdo da seção. Admite apenas atributos globais. Uma página pode conter vários elementos *header*, e estes, por sua vez, os elementos h1-h6.

## Exemplificando

```
<body>
  <header>
    <h1>Aprendendo HTML 5 </h1>
  </header>
</body>
```

▪ **hgroup:** destina-se a agrupar elementos h1-h6 quando existir mais de um deles. Admite apenas atributos globais.

## Exemplificando

```
<hgroup>
  <h1>Título1</h1>
  <h2>Título2</h2>
</hgroup>
```



# Novidades HTML 5

- **keygen:** serve para a manipulação de um par de chaves criptografadas, públicas e privadas para autenticação da comunicação com o servidor por ocasião do envio de formulários.
- **mark:** destina-se a marcar uma palavra ou trecho de texto que deva ser destacado com o propósito de servir de referência.
- **meter:** serve para marcar uma medida escalar compreendida entre determinados limites conhecidos.

## Exemplificando

`<meter value="0.5"></meter>` //0.5 em uma faixa de 0 a 1. Ou seja, 50%.

- **nav:** serve para marcar uma seção da página que contenha links para outras páginas ou para outras partes da própria página. Admite apenas atributos globais.

# Novidades HTML 5

- **output:** serve de container para o resultado de um cálculo. Admite os atributos globais for, form e name.
- **progress:** marca o andamento ou progresso de uma tarefa.
- **ruby:** destina-se a marcar pequenas anotações sobre textos, contendo informações sobre a maneira correta de pronúncia do texto.
- **section:** marca uma seção na página. Podemos chamar de seção um agrupamento de conteúdos tratando de um mesmo tema e, tipicamente contendo um cabeçalho. Admite apenas atributos globais.

## Exemplificando

```
<section>  
  <h1>Elemento Section</h1>  
  <p>Marca uma seção na página</p>  
</section>
```

## Questão 14

**CS UFG 2014 – UEAP - Analista de Tecnologia da Informação – Des. de Sistemas**

Qual é o elemento da linguagem HTML5 que ajuda a adicionar estrutura a um documento e que agrupa, no nível de bloco, itens relacionados com base em seus significados ou em suas funções e não na sua aparência ou leiaute?

a <table>

b <header>

c <div>

d <span>


## Questão 14

**CS UFG 2014 – UEAP - Analista de Tecnologia da Informação – Des. de Sistemas**

Qual é o elemento da linguagem HTML5 que ajuda a adicionar estrutura a um documento e que agrupa, no nível de bloco, itens relacionados com base em seus significados ou em suas funções e não na sua aparência ou leitura?

a <table>

b <header>

 c <div>

d <span>

## Questão 15

### **CESPE 2013 - TRE-MS - Técnico Judiciário - Programação de Sistemas**

Na versão HTML5, novos tipos de campos foram criados. Assinale a opção que descreve corretamente o tipo de campo em HTML.

- a O campo de email, com formatação e validação, dependendo do agente do usuário promove a interação com a agenda de contatos.
- b URL é o campo para endereço web com validação e busca interna de acordo com a URL.
- c O campo tipo telefone (tel) possui máscara de formatação e validação.
- d O campo tipo datetime-local trata automaticamente os horários absolutos de fusos horários de acordo com o servidor da aplicação.
- e O tipo Search é um campo de busca que utiliza o padrão Google em todas as plataformas.

## Questão 15

### **CESPE 2013 - TRE-MS - Técnico Judiciário - Programação de Sistemas**

Na versão HTML5, novos tipos de campos foram criados. Assinale a opção que descreve corretamente o tipo de campo em HTML.

- ➡ a O campo de email, com formatação e validação, dependendo do agente do usuário promove a interação com a agenda de contatos.
- b URL é o campo para endereço web com validação e busca interna de acordo com a URL.
- c O campo tipo telefone (tel) possui máscara de formatação e validação.
- d O campo tipo datetime-local trata automaticamente os horários absolutos de fusos horários de acordo com o servidor da aplicação.
- e O tipo Search é um campo de busca que utiliza o padrão Google em todas as plataformas.

## Questão 16

### FMP-RS 2013 - MPE-AC - Analista - Tecnologia da Informação

A linguagem HTML em sua versão 5 apresenta diversas novas características em relação às versões anteriores dando uma maior potencialidade à linguagem. Dentre as características abaixo qual delas NÃO é advinda da proposta de HTML5?

a Implementação padrão em todos os browsers e com funcionalidade também padrão e consistente em todos.

b Define uma forma padrão de incorporar vídeo e áudio.

c Introdução de diversas APIs, entre elas a API Canvas que permite criar gráficos, - desenhos, jogos e aplicações web.

d Fornece maior interatividade sem a necessidade de Plugins.

e Define novos tipos para o atributo type do comando INPUT usados em formulários, tais como data/hora, números, autofocus e faixa de valores, entre outros.

## Questão 16

### FMP-RS 2013 - MPE-AC - Analista - Tecnologia da Informação

A linguagem HTML em sua versão 5 apresenta diversas novas características em relação às versões anteriores dando uma maior potencialidade à linguagem. Dentre as características abaixo qual delas NÃO é advinda da proposta de HTML5?

- ➡ a Implementação padrão em todos os browsers e com funcionalidade também padrão e consistente em todos.
- b Define uma forma padrão de incorporar vídeo e áudio.
- c Introdução de diversas APIs, entre elas a API Canvas que permite criar gráficos, - desenhos, jogos e aplicações web.
- d Fornece maior interatividade sem a necessidade de Plugins.
- e Define novos tipos para o atributo type do comando INPUT usados em formulários, tais como data/hora, números, autofocus e faixa de valores, entre outros.



## Questão 17

**CS UFG – 2014 - UEAP - Analista de Tecnologia da Informação - Desenvolvimento de Sistemas**

Vários elementos da linguagem HTML 4.01 foram reescritos na nova versão HTML5, ou removidos desta. Qual dos elementos a seguir deixou de ser suportado pela linguagem HTML5?

a <video>

b <table>

c <embed>

d <applet>

## Questão 17

**CS UFG – 2014 - UEAP - Analista de Tecnologia da Informação - Desenvolvimento de Sistemas**

Vários elementos da linguagem HTML 4.01 foram reescritos na nova versão HTML5, ou removidos desta. Qual dos elementos a seguir deixou de ser suportado pela linguagem HTML5?

a <video>

b <table>

c <embed>

➡ d <applet>

## Questão 18

**CESPE 2013 – STF - Técnico Judiciário - Tecnologia da Informação**

A respeito de HTML, XML e CSS3, julgue os itens subsequentes.

HTML5 é uma linguagem multiplataforma que permitiu o desenvolvimento de aplicações web executadas off-line e incorporou novos elementos de conteúdo específico, como <article>, <footer>, <nav> e <section>.

Certo

Errado

## Questão 18

**CESPE 2013 – STF - Técnico Judiciário - Tecnologia da Informação**

A respeito de HTML, XML e CSS3, julgue os itens subsequentes.

HTML5 é uma linguagem multiplataforma que permitiu o desenvolvimento de aplicações web executadas off-line e incorporou novos elementos de conteúdo específico, como <article>, <footer>, <nav> e <section>.

 Certo

Errado

## Questão 19

**FUNDEP 2014 - IFN-MG - Analista de Tecnologia da Informação**

Entre os valores abaixo, IDENTIFIQUE o que NÃO é permitido para o atributo type na versão mais recente da linguagem HTML.

a Tel.

b Email

c Search.

d Data.

## Questão 19

**FUNDEP 2014 - IFN-MG - Analista de Tecnologia da Informação**

Entre os valores abaixo, IDENTIFIQUE o que NÃO é permitido para o atributo type na versão mais recente da linguagem HTML.

a Tel.

b Email

c Search.

 d Data.

## Questão 20

**CESPE 2014 – ANTAQ - Analista Administrativo - Sistemas e Negócios**

Acerca da HTML5, julgue o item subsequente.

A utilização simultânea dos atributos multiple e required no elemento input permite que o usuário especifique mais de um valor, devendo inserir ao menos um valor.

Certo

Errado

## Questão 20

**CESPE 2014 – ANTAQ - Analista Administrativo - Sistemas e Negócios**

Acerca da HTML5, julgue o item subsequente.

A utilização simultânea dos atributos multiple e required no elemento input permite que o usuário especifique mais de um valor, devendo inserir ao menos um valor.

 Certo

Errado



# Questão 21

**FCC 2011 - TRE-PE - Analista Judiciário - Análise de Sistemas**

Na HTML5, o atributo do elemento <input> que especifica uma expressão regular Javascript para validar a entrada de dados em um campo texto é o

a masked.

b expression.

c override.

d placeholder.

e pattern.

## Questão 21

**FCC 2011 - TRE-PE - Analista Judiciário - Análise de Sistemas**


Na HTML5, o atributo do elemento <input> que especifica uma expressão regular Javascript para validar a entrada de dados em um campo texto é o

a masked.

b expression.

c override.

d placeholder.

 e pattern.

## Questão 22

**FCC 2012 - MPE-AP - Técnico Ministerial - Informática**

A linguagem HTML5 traz um conjunto de novos valores possíveis para o atributo type do elemento <input>, com objetivo de definir melhor os tipos de dados de entrada nos campos dos formulários. Fazem parte desses novos valores: tel, url, email, date, time, number,

a char, search, range, meter, color, week e month.

b datetime, datetime-local, string, search e color.

c search, datetime, month, week, datetime-local, range e color.

d date-time, range, color, month, week e boolean.

e datetime, month, week, meter, range, search, boolean e color.

## Questão 22

**FCC 2012 - MPE-AP - Técnico Ministerial - Informática**

A linguagem HTML5 traz um conjunto de novos valores possíveis para o atributo type do elemento <input>, com objetivo de definir melhor os tipos de dados de entrada nos campos dos formulários. Fazem parte desses novos valores: tel, url, email, date, time, number,

a char, search, range, meter, color, week e month.

b datetime, datetime-local, string, search e color.

➡ c search, datetime, month, week, datetime-local, range e color.

d date-time, range, color, month, week e boolean.

e datetime, month, week, meter, range, search, boolean e color.

## Questão 23

**FCC 2012 - MPE-PE - Analista Ministerial - Informática**

Na HTML5, é um atributo booleano que, quando especificado, indica que o conteúdo do elemento <iframe> deve ser renderizado de maneira que pareça fazer parte da página que contém o <iframe> sem bordas ou barras de rolagem.

O atributo citado no texto acima é o:

a smooth.

b pattern.

c seamless.

d sandbox.

e srcdoc.

## Questão 23

**FCC 2012 - MPE-PE - Analista Ministerial - Informática**

Na HTML5, é um atributo booleano que, quando especificado, indica que o conteúdo do elemento <iframe> deve ser renderizado de maneira que pareça fazer parte da página que contém o <iframe> sem bordas ou barras de rolagem.

O atributo citado no texto acima é o:

a smooth.

b pattern.

➡ c seamless.

d sandbox.

e srcdoc.

## Questão 24

### FCC 2014 – TJ-AP - Analista Judiciário - Tecnologia da Informação

Em uma página construída com HTML5 foi criado o seguinte formulário:

```
<form method="post" action="controle.jsp">
<p> <label> Nome <input type="text" name="nome"> </label> </p>
<p> <label> CPF <input type="text" name="CPF"> </label> </p>
<p> <input type="submit" value="Enviar"> </p>
</form>
```

Considere que este formulário foi carregado em um navegador com suporte adequado a HTML5, que os campos foram preenchidos e que o botão “Enviar” foi clicado.

Para que o formulário só seja submetido se o campo CPF possuir 11 números inteiros de 0 a 9 sem pontuação, é necessário colocar na tag input deste campo o atributo de validação

- a) pattern="(0-9)[11]"
- b) pattern="[0-9]{11}"
- c) placeholder="^\d(11)[0-9]\$"
- d) placeholder="(11)[0-9]"
- e) validation="(0-9)[11]"

## Questão 24

### FCC 2014 – TJ-AP - Analista Judiciário - Tecnologia da Informação

Em uma página construída com HTML5 foi criado o seguinte formulário:

```
<form method="post" action="controle.jsp">
<p> <label> Nome <input type="text" name="nome"> </label> </p>
<p> <label> CPF <input type="text" name="CPF"> </label> </p>
<p> <input type="submit" value="Enviar"> </p>
</form>
```

Considere que este formulário foi carregado em um navegador com suporte adequado a HTML5, que os campos foram preenchidos e que o botão “Enviar” foi clicado.

Para que o formulário só seja submetido se o campo CPF possuir 11 números inteiros de 0 a 9 sem pontuação, é necessário colocar na tag input deste campo o atributo de validação

- a) pattern="(0-9)[11]"
- ➔ b) pattern="[0-9]{11}"
- c) placeholder="^\d(11)[0-9]\$"
- d) placeholder="(11)[0-9]"
- e) validation="(0-9)[11]"



## Questão 25

### CESPE 2013 – BACEN - Analista - Análise e Desenvolvimento de Sistemas

Julgue o item subsecutivo, referente ao desenvolvimento de sistemas web HTML/HTML5 e CSS3.

O exemplo abaixo está escrito corretamente em HTML 5 e o atributo required obriga o usuário a escolher uma cor para que o formulário seja enviado.

```
<label>Cor:  
<select name=cor required>  
<option value="">Escolha um  
<option>Vermelho  
<option>Verde  
<option>Azul  
</select>  
</label>
```

Certo

Errado

## Questão 25

### CESPE 2013 – BACEN - Analista - Análise e Desenvolvimento de Sistemas

Julgue o item subsecutivo, referente ao desenvolvimento de sistemas web HTML/HTML5 e CSS3.

O exemplo abaixo está escrito corretamente em HTML 5 e o atributo required obriga o usuário a escolher uma cor para que o formulário seja enviado.

```
<label>Cor:
<select name=cor required>
<option value="">Escolha um
<option>Vermelho
<option>Verde
<option>Azul
</select>
</label>
```

 Certo

Errado

## Questão 26

**CESPE 2014 - TJ-SE - Analista Judiciário - Suporte Técnico em Infraestrutura**

Com relação às formatações de dados HTML, XML e IBM 3270, julgue os itens subsequentes.

Na linguagem HTML5, o uso de prefetch em um elemento <link>, como em <link rel="prefetch" href="/images/image1.jpeg">, permite ao navegador buscar e carregar, em segundo plano, documentos que provavelmente serão chamados a seguir na navegação do usuário.

Certo

Errado

## Questão 26

**CESPE 2014 - TJ-SE - Analista Judiciário - Suporte Técnico em Infraestrutura**

Com relação às formatações de dados HTML, XML e IBM 3270, julgue os itens subsequentes.

Na linguagem HTML5, o uso de prefetch em um elemento <link>, como em <link rel="prefetch" href="/images/image1.jpeg">, permite ao navegador buscar e carregar, em segundo plano, documentos que provavelmente serão chamados a seguir na navegação do usuário.

 Certo

Errado

# Canvas

## Definição

➤ Novo elemento HTML que destina-se a delimitar uma área para criação dinâmica de imagens, tais como, gráficos estáticos, jogos e gráficos dinâmicos e imagens em geral criadas com linguagem de programação dinâmica. Trabalha em conjunto com javascript.

**1) O primeiro passo é definirmos a área de criação, como no exemplo abaixo:**

```
<canvas id="quadro" width="250" height="150"  
  <p>aqui entrará algum conteúdo para navegadores que não suportam canvas.</p>  
</canvas>
```

**2) Agora devemos criar uma referência javascript para o elemento canvas e definir o contexto do desenho.**

```
var elementocanvas = document.getElementById('quadro');  
var ctx = elementocanvas.getContext('2d');
```

# Canvas

## Exemplo Completo

```
<!DOCTYPE html>
<html>
  <body>
    <canvas id="meuCanvas" width="300" height="200" style="border:1px solid #c3c3c3;">
      Seu navegador não suporta a tag <canvas>
    </canvas>
    <script type="text/javascript">
      var canvasElem = document.getElementById('meuCanvas');
      //Obtém o contexto "2d" de desenho para o canvas
      var context = canvasElem.getContext('2d');

      //Desenha retângulo opaco
      context.fillStyle = "rgb(200, 0, 0)";
      context.fillRect(0, 0, 150, 100);

      //Desenha retângulo com transparência
      context.fillStyle = "rgba(0, 0, 200, 0.5)";
      context.fillRect (120, 70, 70, 70);
    </script>
  </body>
</html>
```



# SVG

## Definição

➤ SVG é a abreviatura de **Scalable Vectorial Graphics**, que traduzido para o português significa gráficos vetoriais escaláveis. SVG é uma linguagem XML para descrever gráficos bidimensionais sejam eles de forma dinâmica, estática ou animada. Seu diferencial em relação aos outros formatos vetoriais é de ser um formato aberto, não pertencendo a nenhuma empresa. Ele foi originalmente criado pelo W3C, responsável pela definição de outros padrões como HTML, XHTML, XML.

## Exemplificando

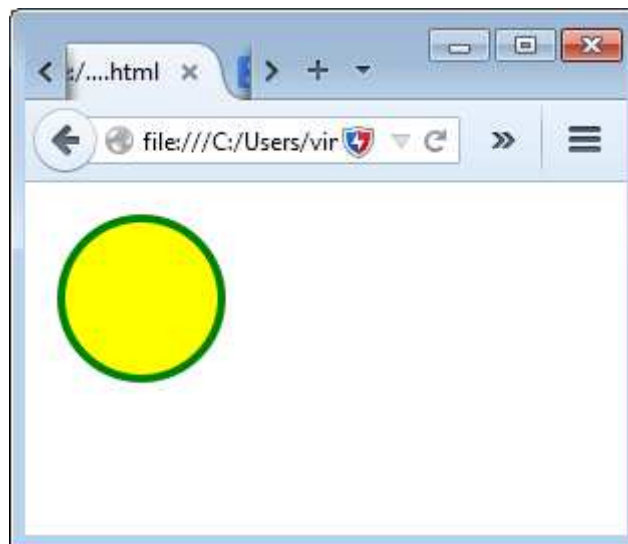
### Desenhando um círculo

```
<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />  
</svg>
```



# SVG

## Resultado do Exemplo Anterior



## Desenhando um retângulo

```
<svg width="400" height="100">  
  <rect width="400" height="100" style="fill:rgb(0,0,255);stroke-width:10;stroke:rgb(0,0,0)" />  
</svg>
```

# SVG

## Resultado do Exemplo Anterior



➤ O SVG é conhecido como um modelo de elementos gráficos de **modo retido** persistindo em um modelo na memória. Análogo ao HTML, o SVG cria um modelo de objeto de elementos, atributos e estilos. Quando o elemento <svg> aparece em um documento HTML5, ele se comporta como um bloco alinhado e faz parte da árvore do documento HTML.

# SVG x Canvas

➤ **SVG** é uma linguagem para descrever gráficos 2D em XML, enquanto Canvas desenha gráficos 2D, assim com JavaScript.

➤ **SVG** é baseado em XML, o que significa que cada elemento está disponível dentro do SVG DOM. Você pode anexar manipuladores de eventos JavaScript para um elemento. Em SVG, cada forma desenhada é manipulada como um objeto. Se os atributos de um objeto SVG são alterados, o navegador pode automaticamente retornar a forma.



➤ **Canvas** é processado pixel por pixel. Na tela, uma vez que o gráfico é desenhado, é “esquecido” pelo browser. Se a sua posição precisar ser mudada, toda a **cena** precisará ser redesenhado, incluindo quaisquer objetos que possam ter sido abrangidos pelo gráfico.

# SVG x Canvas

CANVAS	SVG
Resolução dependente	Resolução independente
Não fornece suporte para manipuladores de eventos	Fornece suporte para manipuladores de eventos
Pouca capacidade de renderização de texto	Adequado para aplicações com grande capacidade de renderização (google maps)
Possibilidade de salvamento da imagem como .png ou .jpg	Renderização pode ficar lenta com objetos complexos (algo que faça uso excessivo de DOM, por exemplo)
Recomendado para jogos que possuam gráficos elaborados	Não recomendado para jogos

## Questão 27 – 01/02

### FCC 2015 – CNMP - Analista de Tecnologia da Informação e Comunicação

Uma página *web*, quando interpretada em um navegador com suporte adequado à linguagem SVG, gerou a seguinte imagem:



Esta página contém o seguinte código-fonte:

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

    </svg>
  </body>
</html>
```

## Questão 27 – 02/02

### FCC 2015 – CNMP - Analista de Tecnologia da Informação e Comunicação

Para gerar a imagem apresentada, entre as tags `<svg>` e `</svg>` deve existir a instrução

a) `< line coords="0,40 40,40 40,80 80,80 80,120 120,120" shape="fill:black; stroke-width:15" />`

b) `< polyline coords="0,40 40,40 40,80 80,80 80,120 120,120" style="fill:white; stroke:black; stroke-width:10" />`

c) `< polyline points="0,40;40,40;40,80;80,80;80,120;120,120" style="stroke:black; stroke-width:15" />`

d) `< line coords="0,40;40,40;40,80;80,80;80,120;120,120" style="stroke:black; stroke-width:15" />`

e) `< polyline points="0,40 40,40 40,80 80,80 80,120 120,120" style="fill:white; stroke:black; stroke-width:10" />`

## Questão 27

### FCC 2015 – CNMP - Analista de Tecnologia da Informação e Comunicação

Para gerar a imagem apresentada, entre as tags `<svg>` e `</svg>` deve existir a instrução

a) `< line coords="0,40 40,40 40,80 80,80 80,120 120,120" shape="fill:black; stroke-width:15" />`

b) `< polyline coords="0,40 40,40 40,80 80,80 80,120 120,120" style="fill:white; stroke:black; stroke-width:10" />`

c) `< polyline points="0,40;40,40;40,80;80,80;80,120;120,120" style="stroke:black; stroke-width:15" />`

d) `< line coords="0,40;40,40;40,80;80,80;80,120;120,120" style="stroke:black; stroke-width:15" />`

➡ e) `< polyline points="0,40 40,40 40,80 80,80 80,120 120,120" style="fill:white; stroke:black; stroke-width:10" />`

## Questão 28

### CESPE 2013 – MPOG - Tecnologia da Informação

A respeito de interface web, ferramentas de desenho e padrões web, julgue os próximos itens.

Elementos em formato SVG podem ser embutidos diretamente em uma página HTML usando HTML5.

Certo

Errado



## Questão 28

### CESPE 2013 – MPOG - Tecnologia da Informação

A respeito de interface web, ferramentas de desenho e padrões web, julgue os próximos itens.

Elementos em formato SVG podem ser embutidos diretamente em uma página HTML usando HTML5.

➡ Certo

Errado

## Questão 29

### FJG 2014 - Câmara Municipal do Rio – Analista Legislativo - Desenvolvimento

Atualmente, HTML5 representa a versão atual da HTML, utilizada como uma linguagem interpretada na construção de páginas para sites na internet.

Na HTML5, emprega-se, para desenhar gráficos via scripting, normalmente JavaScript, por meio do método `getContext()`, a seguinte tag:

- (A) `<draw>`
- (B) `<canvas>`
- (C) `<design>`
- (D) `<graphics>`

## Questão 29

### FJG 2014 - Câmara Municipal do Rio – Analista Legislativo - Desenvolvimento

Atualmente, HTML5 representa a versão atual da HTML, utilizada como uma linguagem interpretada na construção de páginas para sites na internet.

Na HTML5, emprega-se, para desenhar gráficos via scripting, normalmente JavaScript, por meio do método `getContext()`, a seguinte tag:

(A) `<draw>`

➡ (B) `<canvas>`

(C) `<design>`

(D) `<graphics>`

## Questão 30

FCC 2014 - TRF - 4ª Região - Analista Judiciário - Informática

O elemento <canvas> da linguagem HTML5 é utilizado para renderizar gráficos bitmap dinâmicos em uma página, permitindo a criação de jogos e animações, utilizando JavaScript.

O fragmento de código da página HTML a seguir mostra um exemplo de uso da API Canvas, que gera um retângulo de cor cinza na tela.

Para que o exemplo funcione corretamente, as lacunas I e II devem ser preenchidas, respectivamente, por

- a ret e HTTPCanvas.
- b ret e 2d.
- c root e 3d.
- d ret e root.
- e canvas e 3d.

```
<body>
  <canvas id="ret" width="150" height="150"></canvas>
  <script type="text/javascript">
    var c = document.getElementById(' I ');

    var contexto = c.getContext(' II ');

    contexto.fillStyle = '#C0C0C0';
    contexto.fillRect(10, 20, 180, 100);
  </script>
</body>
```

## Questão 30

FCC 2014 - TRF - 4ª Região - Analista Judiciário - Informática

O elemento <canvas> da linguagem HTML5 é utilizado para renderizar gráficos bitmap dinâmicos em uma página, permitindo a criação de jogos e animações, utilizando JavaScript.

O fragmento de código da página HTML a seguir mostra um exemplo de uso da API Canvas, que gera um retângulo de cor cinza na tela.

Para que o exemplo funcione corretamente, as lacunas I e II devem ser preenchidas, respectivamente, por

```
<body>
  <canvas id="ret" width="150" height="150"></canvas>
  <script type="text/javascript">
    var c = document.getElementById(' I ');
    var contexto = c.getContext(' II ');
    contexto.fillStyle = '#C0C0C0';
    contexto.fillRect(10, 20, 180, 100);
  </script>
</body>
```

a ret e HTTPCanvas.

➔ b ret e 2d.

c root e 3d.

d ret e root.

e canvas e 3d.

## Questão 31

### FCC 2012 - TRE-SP - Técnico Judiciário - Programação de Sistemas

A linguagem HTML5 permite a utilização de uma série de APIs que ajudam na criação de aplicações Web. Estas APIs podem ser acessadas por meio de ..I.. e usadas em conjunto com alguns elementos HTML. A API ..II.. permite desenhar na tela do navegador utilizando o elemento ..III..e..IV..

As lacunas I, II, III e IV são preenchidas correta e respectivamente por:

- a tags XML, Canvas, <SVG>, XML.
- b linguagens dinâmicas, Scalable Vector Graphics (SVG), <canvas>,XML.
- c scripts JavaScript, Canvas, <canvas>, JavaScript.
- d tags Scalable Vector Graphics (SVG), SVG, <canvas>, JavaScript.
- e taglibs Java, Canvas, <canvas>, XML.

## Questão 31

### FCC 2012 - TRE-SP - Técnico Judiciário - Programação de Sistemas

A linguagem HTML5 permite a utilização de uma série de APIs que ajudam na criação de aplicações Web. Estas APIs podem ser acessadas por meio de ..I.. e usadas em conjunto com alguns elementos HTML. A API ..II.. permite desenhar na tela do navegador utilizando o elemento ..III..e..IV..

As lacunas I, II, III e IV são preenchidas correta e respectivamente por:

a tags XML, Canvas, <SVG>, XML.

b linguagens dinâmicas, Scalable Vector Graphics (SVG), <canvas>,XML.

➡ c scripts JavaScript, Canvas, <canvas>, JavaScript.

d tags Scalable Vector Graphics (SVG), SVG, <canvas>, JavaScript.

e taglibs Java, Canvas, <canvas>, XML.

# HTML Multimídia

## Definição

- A definição genérica sobre multimídia é: quase qualquer coisa que você pode ouvir ou ver.

## Exemplos

- Fotos, música, som, vídeos, discos, filmes, animações e muito mais. Páginas da Web, muitas vezes contém elementos multimídia de diferentes tipos e formatos.

## Compatibilidade de Navegador

- Os primeiros navegadores tinham suporte para apenas texto, limitado a uma única fonte, em uma única cor. Mais tarde vieram navegadores com suporte para cores e fontes, e até mesmo suporte para fotos!
- O suporte para sons, animações e vídeos é tratado de forma diferente por vários navegadores. Diferentes tipos e formatos são suportados, e alguns formatos exigem programas auxiliares extras (plugins) para funcionarem adequadamente. HTML5 multimídia promete um futuro mais fácil para multimídia.



# HTML Multimídia

- Elementos multimídia (como sons ou vídeos) são armazenados em arquivos de mídia.
- A maneira mais comum para descobrir o tipo de um arquivo é olhar para a extensão do mesmo. Quando um navegador vê o arquivo de extensão .htm ou .html, ele vai tratar o arquivo como um arquivo HTML. A extensão .xml indica um arquivo XML, e a extensão .css indica um arquivo de folha de estilo. Imagens são reconhecidas por extensões como .gif, .png e .jpg.
- Arquivos multimídia também têm seus próprios formatos e diferentes extensões como: .swf, .wav, .mp3, .mp4, .mpg, .wmv, e .avi.

## Formatos de Vídeo

- MP4 é o próximo formato padrão para vídeos na internet. Além de ser o formato recomendado pelo YouTube, ele é suportado por “flash players”.



# HTML Multimídia

FORMATO	EXTENSÃO	DESCRIÇÃO
MPEG	.mpg, .mpeg	Desenvolvido pelo Moving Pictures Expert Group, sendo o primeiro formato de vídeo popular na web. Usado para ser suportado por todos os navegadores, mas não é suportado em HTML5 (substituído por MP4).
AVI	.avi	O Audio Video Interleave foi desenvolvido pela Microsoft, e é comumente usado em câmeras de vídeo e hardwares de TV.
WMV	.wmv	O Windows Media Video possui características similares ao AVI.
QuickTime	.mov	Desenvolvido pela Apple, é comumente usado em câmeras de vídeo e hardwares de TV.
RealVideo	.rm, .ram	Desenvolvido pela Real Media para permitir streaming de vídeo com baixas larguras de banda. Ainda é usado para o vídeo on-line e não funciona bem navegadores web.

# HTML Multimídia

FORMATO	EXTENSÃO	DESCRIÇÃO
Flash	.swf, .flv	Desenvolvido pela Macromedia. Requer um plug-in para funcionar em navegadores web.
Ogg	.ogg	Desenvolvido pela Fundação Xiph.Org. É compatível com HTML5.
WebM	.webm	Desenvolvido pelos gigantes da Internet (Mozilla, Opera, Adobe e Google). É compatível com HTML5.
MPEG-4 ou MP4	.mp4	Desenvolvido pelo Moving Pictures Expert Group com base no QuickTime. É comumente usado em câmeras de vídeo mais recentes e TVs. É suportado por todos os navegadores HTML5 e é recomendado pelo YouTube.

**IMPORTANTE:** Apenas MP4, WebM e Ogg são suportados em HTML5.

# HTML Multimídia

## Formatos de Som

➤ MP3 é o formato padrão para músicas na internet e o termo MP3 tornou-se sinônimo de música digital. Apenas MP3, WAV e Ogg são suportados em HTML5.

FORMATO	EXTENSÃO	DESCRIÇÃO
MIDI	.midi, .mid	Musical Instrument Digital Interface é o principal formato para os dispositivos da música eletrônica como sintetizadores e placas de som. Os arquivos MIDI não contêm som, mas notas digitais que podem ser reproduzidas por aparelhos eletrônicos. Funciona bem em todos os computadores, mas não em navegadores web.
RealAudio	.rm, .ram	Desenvolvido pela Real Media para permitir streaming de áudio de com baixas larguras de banda. Não funciona em navegadores web.
WMA	.wma	Windows Media Audio foi desenvolvido pela Microsoft e é usado em players música. Funciona bem em computadores com Windows, mas não em navegadores web.

# HTML Multimídia

FORMATO	EXTENSÃO	DESCRIÇÃO
AAC	.aac	Advanced Audio Coding foi desenvolvido pela Apple como o formato padrão para o iTunes. Funciona bem em computadores da Apple, mas não em navegadores web.
MP3	.mp3	Arquivos MP3 são a parte de som de arquivos MPEG. MP3 é o formato mais popular para os players de música. Combina boa compressão (arquivos pequenos) com alta qualidade. Compatível com todos os navegadores.
MP4	.mp4	MP4 é um formato de vídeo, mas pode também ser usado para o áudio. Vídeo MP4 é o próximo formato de vídeo da internet. Isto leva a suporte automático para MP4 áudio por todos os browsers.
WAV	.wav	Desenvolvido pela IBM e Microsoft. Funciona bem em sistemas operacionais Windows, Macintosh e Linux e é Compatível com HTML5.
Ogg	.ogg	Desenvolvido pela Fundação Xiph.Org. É compatível com HTML5.

# Vídeo em HTML5

- Antes de HTML5, não havia padrão para exibição de vídeos em uma página web e os vídeos só poderiam ser reproduzidos através um plugin (como flash).
- A tag HTML5 <video> especifica uma forma padrão de inserir um vídeo em uma página da web.

Internet Explorer 9 +, Firefox, Opera, Chrome, Safari suportam o elemento <video>.

**OBS:** Internet Explorer 8 e versões anteriores não suportam o elemento <video>.

## Exemplificando

```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogv" type="video/ogg">  
  Seu browser não suporta a tag <video>.  
</video>
```

# Vídeo em HTML5

## Como Funciona?

- Os controles atribuem controles de vídeo, como play, pause e volume.
- Texto entre o <video> e </ video> só será exibido em navegadores que não suportam o elemento <video>.
- É uma boa idéia para incluir sempre os atributos largura e altura.
- Se a altura e largura não estão definidas, o navegador não sabe o tamanho do vídeo. Sendo assim, a página será redimensionada enquanto o vídeo é exibido.
- Diversos elementos <source> podem conectar-se a diferentes arquivos de vídeo. O browser usará o primeiro formato reconhecido.



# Vídeo em HTML5

BROWSER	MP4	WEBM	Ogg
IE	SIM	NÃO	NÃO
Chrome	SIM	SIM	SIM
Firefox	SIM	SIM	SIM
Safari	SIM	NÃO	NÃO
Opera	SIM (apenas Opera 25)	SIM	SIM

## HTML Video – Métodos, Propriedades e Eventos

- HTML5 define métodos DOM, propriedades e eventos para o elemento <video>. Isso permite que você carregue, toque e pause vídeos, bem como defina duração e volume.
- Há também eventos DOM que podem notificá-lo quando um vídeo começar a executar, estiver em pausa, etc.



# Áudio em HTML5

- Antes de HTML5, não havia nenhuma norma para a reprodução de arquivos de áudio em uma página web e os arquivos de áudio só poderiam ser tocados com um plug-in (como flash).
- O HTML5 <audio> especifica uma forma padrão para inserir áudio em uma página web.

Internet Explorer 9 +, Firefox, Opera, Chrome, Safari suportam a tag <audio>.

**DICA:** Internet Explorer 8 e versões anteriores, não suportam a tag <audio>.

## Exemplificando

### <audio controls>

```
<source src="musica1.ogg" type="audio/ogg">
```

```
<source src="musica2.mp3" type="audio/mpeg">
```

Seu browser não suporta o elemento <audio>.

```
</audio>
```

# Áudio em HTML5

## Como Funciona?

- Os controles atribuem controles de áudio, como play, pause e volume.
- O texto entre o <audio> e </ audio> é exibido em browsers que não suportam o elemento <audio>.
- Vários elementos <source> podem conectar-se a diferentes arquivos de áudio. O browser usará o primeiro formato reconhecido.



## Suporte do Navegador

- Atualmente, existem três formatos de arquivos suportados pelo elemento <audio>: MP3, WAV e Ogg.

# Áudio em HTML5

BROWSER	MP3	WAV	Ogg
IE	SIM	NÃO	NÃO
Chrome	SIM	SIM	SIM
Firefox	SIM	SIM	SIM
Safari	SIM	SIM	NÃO
Opera	SIM	SIM	SIM

## HTML Audio - Métodos, Propriedades e Eventos

- HTML5 define métodos DOM, propriedades e eventos para o elemento <audio>. Isso permite que você carregue, execute e pause, assim como defina a duração e o volume de áudio.
- Há também eventos DOM que podem notificá-lo quando um áudio começa a tocar, está em pausa, etc.

# Áudio em HTML5

## Tag's para Audio em HTML 5

TAG	DESCRIÇÃO
<audio>	Define um conteúdo sonoro
<source>	Define diversos recursos de mídia, tais como <video> e <audio>

# HTML Helpers (Plugins)

## Definição

- Helpers são componentes que estendem funcionalidades default dos browsers e são também chamados de plugins.
- Exemplos famosos de HTML Helpers são os Applets Java.
- Plugins podem ser adicionados às páginas com as tags **<object>** ou **<embed>** e podem ser utilizados por vários motivos: exibição de mapas, busca por vírus, verificações de segurança bancária, etc.

## Elemento Object

- O elemento **<object>** é suportado por todos os navegadores e ele define um objeto incorporado em um documento HTML.
- O elemento **<object>** é usado para incorporar plugins (como applets, leitores de PDF e flash players) em páginas da web.

# HTML Helpers (Plugins)

## Exemplificando

```
<object width = "400" height = "50" dados = "bookmark.swf"> </ object>
```

**O elemento <object> também pode ser usado para incluir HTML em HTML:**

```
<object width = "100%" height = "500px" dados = "snippet.html"> </ object>
```

## O Elemento <embed>

- O elemento <embed> é suportado em todos os principais navegadores e também define um objeto incorporado em um documento HTML.
- Os navegadores têm apoiado o elemento <embed> por um longo tempo. No entanto, ele não fez parte da especificação HTML antes de HTML 5. O elemento irá funcionar em uma página HTML5, mas não em uma página HTML 4.

# HTML Helpers (Plugins)

## Exemplificando

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

➤ Note que o elemento `<embed>` não tem uma tag de fechamento. Logo, ele não pode conter um texto alternativo.

**O elemento `<embed>` também pode ser usado para incluir HTML em HTML:**

```
<embed width="100%" height="500px" src="snippet.html">
```

**Também serve para incluir imagens:**

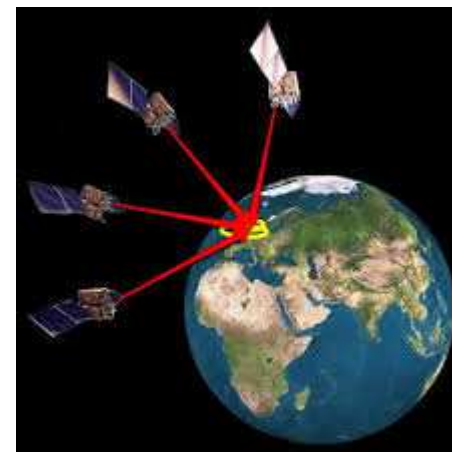
```
<embed src = "audi.jpeg">
```



# Geolocalização

## Definição

- Geolocalização é usada para localizar a posição de um usuário. Ela oferece acesso via script à localização geográfica do dispositivo do usuário que hospeda a funcionalidade.
- Uma vez que esta pode comprometer a privacidade do usuário, a posição não está disponível, a menos que o usuário a aprove.
- Segundo as especificações da API, nenhuma garantia é fornecida de que os dados retornados pelas funcionalidades da API forneçam a localização exata do usuário.



## SUORTE

Internet Explorer 9 ou superior, Firefox, Chrome, Safari e Opera suportam geolocalização.

**OBS:** Geolocalização é muito mais precisa para dispositivos com GPS, como o iPhone.



# Geolocalização

## Como Funciona?

- Devemos usar o método **getCurrentPosition ()** para obter a posição do usuário. O exemplo a seguir é um exemplo de Geolocation simples que obtêm a latitude e a longitude da posição do usuário:

```
<script>
var x = document.getElementById("demo");
function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(showPosition);
    } else {
        x.innerHTML = "Geolocation is not supported by this browser.";
    }
}
function showPosition(position) {
    x.innerHTML = "Latitude: " + position.coords.latitude +
    "<br>Longitude: " + position.coords.longitude;
}
</script>
```

# Geolocalização

## Explicando o Exemplo:

- 1) É verificado se a geolocalização é suportada.
- 2) Se suportada, é invocado o método `getCurrentPosition ()`.
- 3) Se não, será exibida uma mensagem para o usuário.
- 4) Se o método `getCurrentPosition ()` é executado com sucesso, ele retorna as coordenadas para a função especificada no parâmetro (`showPosition`).
- 5) A função `showPosition ()` recebe a Latitude e Longitude.

No exemplo a seguir, veremos o tratamento de erros em geolocalização:

# Geolocalização

## Tratamento de Erros

```
function showError(error) {  
    switch(error.code) {  
        case error.PERMISSION_DENIED:  
            x.innerHTML = "Permissão negada."  
            break;  
        case error.POSITION_UNAVAILABLE:  
            x.innerHTML = "Informação de localização indisponível."  
            break;  
        case error.TIMEOUT:  
            x.innerHTML = "Tempo de resposta expirado."  
            break;  
        case error.UNKNOWN_ERROR:  
            x.innerHTML = "Erro desconhecido."  
            break;  
    }  
}
```

# Geolocalização

## Códigos de Erro

Permission denied - O usuário não permitiu geolocalização.

Position unavailable - Não é possível obter a localização atual.

Timeout - A operação expirou.

## Exibindo o Resultado em um Mapa

Para visualizar o resultado em um mapa, você precisa ter acesso a um serviço de mapas que pode usar latitude e longitude, como o Google Maps:

## Exemplificando

```
function showPosition(position) {  
    var latlon = position.coords.latitude + "," + position.coords.longitude;  
    var img_url = "http://maps.googleapis.com/maps/api/staticmap?center=  
    "+latlon+"&zoom=14&size=400x300&sensor=false";  
    document.getElementById("mapholder").innerHTML = "<img src='"+img_url+"'>";  
}
```

# Geolocalização

➤ No exemplo anterior, usamos os dados de latitude e longitude para exibir a localização em um mapa do google (usando uma imagem estática).

## **Informações Sobre uma Localização Específica**

➤ Esta página demonstrou como exibir a posição de um usuário em um mapa. No entanto, a geolocalização também é muito útil para obter informações específicas do local, tais como:

- **Atualizar informações locais**
- **Exibir locais de interesse próximos do usuário**
- **Dados de navegação (GPS)**

➤ O método `getCurrentPosition()` retorna um objeto se a execução for bem-sucedida. As propriedades de latitude, longitude e precisão são sempre retornadas. As demais propriedades a seguir são retornadas, se disponíveis.

# Geolocalização

PROPRIEDADE	DESCRIÇÃO
coords.latitude	A latitude como um número decimal.
coords.longitude	A longitude como um número decimal.
coords.accuracy	A posição precisa.
coords.altitude	A altitude em metros acima do nível do mar.
coords.altitudeAccuracy	A altitude precisa.
coords.heading	A posição em graus no sentido horário a partir do Norte
coords.speed	A velocidade em metros por segundo.
timestamp	A data/hora da resposta.

# Geolocalização

## Outros Métodos Importantes

**watchPosition()** - Retorna a posição atual do usuário e continua a retornar posição atualizada à medida que o usuário se move (como o GPS em um carro).

**clearWatch()** - Interrompe o método watchPosition().

O exemplo a seguir mostra o método watchPosition(). É necessário um dispositivo GPS precisa de testar isso (como o iPhone).

```
<script>
  var x = document.getElementById("demo");
  function getLocation() {
    if (navigator.geolocation) {
      navigator.geolocation.watchPosition(showPosition);
    } else {
      x.innerHTML = "Geolocation is not supported by this browser.";
    }
  }
  function showPosition(position) {
    x.innerHTML = "Latitude: " + position.coords.latitude +
      "<br>Longitude: " + position.coords.longitude;
  }
</script>
```

# HTML 5 Drag and Drop

➤ A API de Drag and Drop é relativamente simples. Basicamente, inserir o atributo draggable="true" num elemento o torna arrastável. E há uma série de eventos que você pode tratar.

## Os eventos do objeto sendo arrastado são:

- **dragstart:** o objeto começou a ser arrastado. O evento que a função recebe tem um atributo target, que contém o objeto sendo arrastado.
- **drag:** o objeto está sendo arrastado.
- **dragend:** a ação de arrastar terminou.

## O objeto sobre o qual outro é arrastado sofre os seguintes eventos:

- **dragenter:** o objeto sendo arrastado entrou no objeto target.
  - **dragleave:** o objeto sendo arrastado deixou o objeto target
  - **dragover:** o objeto sendo arrastado se move sobre o objeto target
  - **drop:** o objeto sendo arrastado foi solto sobre o objeto target



# HTML 5 Drag and Drop

## EXEMPLO

```
<!DOCTYPE HTML>
<html>
  <head>
    <script>
      function allowDrop(ev) {
        ev.preventDefault();
      }
      function drag(ev) {
        ev.dataTransfer.setData("text", ev.target.id);
      }
      function drop(ev) {
        ev.preventDefault();
        var data = ev.dataTransfer.getData("text");
        ev.target.appendChild(document.getElementById(data));
      }
    </script>
  </head>
  <body>
    <div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>
    
  </body>
</html>
```

# HTML 5 Drag and Drop

➤ Pode parecer complicado, mas vamos passar por todas as diferentes partes de um evento de arrastar e soltar.

## Tornando um Elemento “Arrastável”

Primeiro de tudo: para fazer um elemento “arrastável”, defina o atributo draggable como true:

```
<img draggable = "true">
```

## ONDRAGSTART e SETDATA

Em seguida, especifique *o que deve acontecer quando o elemento é arrastado*.

No exemplo acima, o atributo `ondragstart` chama uma função, de “drag” (evento), que especifica os dados a ser arrastado.

O método `dataTransfer.setData()` define o tipo de dados e o valor dos dados a serem arrastados:

```
function drag(ev) {  
    ev.dataTransfer.setData("text", ev.target.id);  
}
```

Neste caso, o tipo de dados é “text” e o valor é o id do elemento “arrastável” (“drag1”).

# HTML 5 Drag and Drop

## Onde “soltar” o Elemento? - OnDragOver

O evento onDragOver especifica onde os dados arrastados podem ser “soltos”.

Por padrão, os dados/elementos não podem ser “soltos” em outros elementos. Para permitir “soltá-los”, devemos evitar a manipulação padrão do elemento.

Isso é feito chamando o método `event.preventDefault ()` para o evento `OnDragOver`:  
`event.preventDefault ()`

## Fazendo o “drop” - ondrop

Quando os dados arrastados são “soltos”, um evento de “drop” ocorre.

No exemplo acima, o atributo `ondrop` chama uma função `drop` (evento):

```
function drop(ev) {  
    ev.preventDefault();  
    var data = ev.dataTransfer.getData("text");  
    ev.target.appendChild(document.getElementById(data));  
}
```

# HTML 5 Drag and Drop

## Explicando o Código

- Invoque `preventDefault ()` para impedir a manipulação padrão do navegador dos dados (o padrão é aberto como um link “on drop”)
- Obtenha os dados arrastados com o método `dataTransfer.getData ()`. Este método irá devolver quaisquer dados que foram definidos para o mesmo tipo no método `setData()`.
- O dado arrastado é o id do elemento arrastado ("drag1"). Acrescente o elemento arrastado no elemento “solto”.

**SUPORTE:** Internet Explorer 9+, Firefox, Opera, Chrome e Safari suportam Drag and Drop.

Drag and drop não funciona no Safari 5.1.7 e versões anteriores.

# API Storage

## Problema Antigo

➤ Antes de HTML 5, os dados do aplicativo eram armazenados em cookies, incluídos a cada pedido do servidor. Os cookies apresentavam um limite de armazenamento baixo.

## Boa Prática?

➤ O armazenamento local é mais seguro, e grandes quantidades de dados podem ser armazenados localmente, sem afetar o desempenho do site.

## Vantagens da API Storage

- Ao contrário dos cookies, o limite de armazenamento é muito maior (pelo menos 5 MB) e as informações não são transferidas para o servidor.
- O armazenamento local funciona por domínio. Todas as páginas, de um domínio, podem armazenar e acessar os mesmos dados.
- A utilização de HTML5 Local Storage é uma melhor escolha do que o uso de cookies na aplicação.

# API Storage

## Histórico de Sessão

- O objeto history foi acrescido de dois novos métodos em HTML 5:
  - **pushState**(data,title[,url]): acrescenta uma entrada na lista de histórico.
  - **replaceState**(data,title[,url]): modifica a entrada atual na lista de histórico.
- Com isso, você pode acrescentar itens à lista de histórico, associando dados ou mesmo uma URL a eles. Por exemplo, digamos que você tenha três elementos de conteúdo em sua página e um script que exiba um por vez de acordo com os cliques do usuário no menu:

```
function showContent(n){  
    // Escondemos todos os elementos de conteúdo  
    for(var i=1;i<4;i++)  
        document.getElementById('cont'+i).style.display='none'  
    // Exibimos o elemento escolhido  
    document.getElementById('cont'+n).style.display='block'  
}
```

# API Storage

➤ Vamos fazer com que nosso script acrescente uma linha de histórico ao selecionar um elemento:

```
function showPage(n){  
    // Escondemos todos os elementos de conteúdo  
    for(var i=1;i<4;i++)  
        document.getElementById('cont'+i).style.display='none'  
    // Exibimos o elemento escolhido  
    document.getElementById('cont'+n).style.display='block'  
}  
function showContent(n){  
    // Mostramos o conteúdo escolhido  
    showPage(n)  
    // Salvamos a página atual no histórico  
    history.pushState({page:n},'Conteúdo '+n)  
}
```

# API Storage

➤ Fazendo isso, cada vez que o usuário escolher um item no menu, o elemento será exibido e uma linha será acrescentada no histórico. O usuário poderá acessar normalmente esses itens de histórico usando o botão de voltar do navegador. Cada vez que ele usar o histórico, será disparado um evento *popstate*. Assim, para que nosso script esteja completo, basta tratar esse evento:

```
function showPage(n){
    // Escondemos todos os elementos de conteúdo
    for(var i=1;i<4;i++)
        document.getElementById('cont'+i).style.display='none'
    // Exibimos o elemento escolhido
    document.getElementById('cont'+n).style.display='block'
}
function showContent(n){
    // Mostramos o conteúdo escolhido
    showPage(n)
    // Salvamos a página atual no histórico
    history.pushState({page:n},'Conteúdo '+n)
}
// Quando o usuário navegar no histórico, mostramos a página relacionada:
window.onpopstate=function(e){
    if(e.state)
        showPage(e.page)
}
```



# API Storage

## Objeto Local Storage

➤ O objeto `localStorage` armazena os dados sem data de validade. Os dados não serão excluídos quando o navegador é fechado, e estará disponível no próximo dia, semana ou ano.

## Exemplificando

```
// Guardando
localStorage.setItem("ultimonome", "Reis");
// Recuperando
document.getElementById("result").innerHTML = localStorage.getItem("ultimonome");
```

## Explicando o Exemplo

- 1) Criamos um par de chave/valor com nome = "ultimonome" e valor = "reis";
- 2) Recuperamos o valor do "ultimonome" e inserimos no elemento com id = "result";

# API Storage

- O exemplo anterior também poderia ser escrito assim:

```
// Guardando
localStorage.ultimonome = "Reis";
// Recuperando
document.getElementById("result").innerHTML = localStorage.ultimonome;
```

- A sintaxe para remover o "ultimonome" do localStorage é o seguinte:

```
localStorage.removeItem("ultimonome");
```

**IMPORTANTE:** Os pares chave/valor são sempre armazenados como strings. Lembre-se de convertê-los para outro formato quando for necessário!

- O exemplo a seguir conta o número de vezes que um usuário clica em um botão. Neste código o valor da string é convertido em numérico para ser capaz de incrementar o contador.

# API Storage

```
if (localStorage.quantidadedecliques) {  
    localStorage.quantidadedecliques = Number(localStorage.quantidadedecliques) + 1;  
} else {  
    localStorage.quantidadedecliques = 1;  
}
```

```
document.getElementById("result").innerHTML = "Você pressionou o botão " +  
localStorage.quantidadedecliques + " vez(es).";
```

## O Objeto Session Storage

- O objeto sessionStorage é igual ao objeto localStorage, exceto pelo fato do armazenamento dos dados para uma única sessão. Os dados são apagados quando o usuário fecha a janela do navegador.
- O exemplo a seguir conta o número de vezes que um usuário clica em um botão, na sessão atual:

# API Storage

## Exemplificando

```
if (sessionStorage.quantcliques) {  
    sessionStorage.quantcliques = Number(sessionStorage.quantcliques) + 1;  
}  
else {  
    sessionStorage.quantcliques = 1;  
}
```

```
document.getElementById("result").innerHTML = "Você pressionou o botão " +  
sessionStorage.quantcliques + " vez(es) nesta sessão.";
```

**SUPORTE:** Internet Explorer 8+, Firefox, Opera, Chrome e Safari.

Internet Explorer 7 e versões anteriores, não suportam Local Storage.

# Aplicações Offline

➤ Com Application Cache é fácil fazer uma versão offline de uma aplicação web através da criação de um arquivo de manifesto cache.

## Definição

➤ HTML5 introduz application cache, o que significa que uma aplicação web é armazenada em cache e fica acessível sem uma conexão com a internet.

## Vantagens

- Navegação off-line - os usuários podem usar o aplicativo quando estão offline
- Velocidade – os recursos são carregados com mais velocidade.
- Redução de carga do servidor - o navegador só irá baixar do servidor os recursos modificados.

# Aplicações Offline

➤ O exemplo abaixo mostra um documento HTML com um manifesto de cache (para navegação offline):

```
<!DOCTYPE HTML>
<html manifest="demo.appcache">
  <body>
    Conteúdo do Elemento
  </body>
</html>
```

## Dicas Básicas

➤ Para habilitar o cache de aplicativo, deve-se incluir o atributo “manifest” na tag <html> do documento:

➤ Toda página com o atributo de manifesto especificado será armazenada em cache quando o usuário visitar a página. Se o atributo “manifest” não for especificado, a página não será armazenada em cache (a menos que a página seja especificada diretamente no arquivo de manifesto).

**DICA:** A extensão recomendada para arquivos de manifesto é: ".appcache"

# Aplicações Offline

**DICA:** Um arquivo de manifesto deve ser criado com o tipo de mídia correto, que é "text/cache-manifest". Isso deve ser configurado no servidor web.

## O Arquivo de Manifesto

➤ O arquivo de manifesto é um arquivo de texto simples que informa ao navegador o que deverá ser armazenado em cache (e o que não será armazenado).

### O arquivo de manifesto tem três seções:

- CACHE MANIFEST - Os arquivos listados sob este cabeçalho serão armazenados em cache após o primeiro download.
- NETWORK - Os arquivos listados sob este cabeçalho requerem uma conexão com o servidor e nunca serão armazenados em cache.
- FALLBACK - Os arquivos listados sob este cabeçalho especificam páginas fallback de uma página é inacessível

# Aplicações Offline

## Exemplificando

Segue abaixo um exemplo de uso da seção CACHE MANIFEST:

CACHE MANIFEST

/theme.css

/logo.gif

/main.js

➤ O arquivo de manifesto acima enumera três recursos: um arquivo CSS, uma imagem GIF e um arquivo JavaScript. Quando o arquivo de manifesto é carregado, o navegador irá baixar os três arquivos do diretório raiz do site.

A partir daí, sempre que o usuário não estiver conectado à internet, os recursos ainda estarão disponíveis.



# Aplicações Offline

## Exemplificando

Segue abaixo um exemplo de uso da seção NETWORK:

➤ A seção NETWORK abaixo especifica que o arquivo "login.asp" nunca deve ser armazenado em cache e não estará disponível off-line:

NETWORK:  
login.asp

**OBS:** Um asterisco pode ser usado para indicar que todos os outros recursos/arquivos requerem uma conexão com a internet:

NETWORK:  
\*

# Aplicações Offline

## Exemplificando

Segue abaixo um exemplo de uso da seção FALLBACK:

➤ A seção FALLBACK abaixo especifica que "offline.html" será encontrada no local de todos os arquivos no diretório “/html” no caso de não ser possível estabelecer uma conexão com a internet:

FALLBACK:  
/html//offline.html

**OBS:** A primeira URI é o recurso, a segunda é o fallback.

# Aplicações Offline

## Atualizando o Cache

- Depois que um aplicativo é armazenado em cache, ele permanece em cache até que uma das seguintes situações ocorra:
  - O usuário apaga os dados armazenados em cache para a aplicação;
  - O desenvolvedor atualiza o arquivo de manifesto;
  - O desenvolvedor usa o objeto `applicationCache` para verificar o status do cache e atualizá-lo.

# Aplicações Offline

## Exemplificando

### Arquivo CACHE MANIFEST Completo

CACHE MANIFEST

# 2015-04-14 v1.0.0

/theme.css

/logo.gif

/main.js

NETWORK:

login.asp

FALLBACK:

/html/ /offline.html

# Aplicações Offline

**DICA:** As linhas que começam com um "#" são linhas de comentário, mas também podem servir para outra finalidade. Application Cache só é atualizado quando arquivo de manifesto é modificado. Se você editar uma imagem ou mudar uma função JavaScript, essas alterações não irão gerar alterações no cache. Atualizando a data e a versão em uma linha de comentário é uma maneira de fazer a alteração no cache do navegador.

## **Cuidado:**

Tenha cuidado com o que você armazenar em cache. Uma vez que um arquivo é armazenado em cache, o navegador irá continuar a mostrar a versão em cache, mesmo se você alterar o arquivo no servidor. Para garantir a atualização do cache, é necessária a alteração do arquivo de manifesto.

Os navegadores podem ter diferentes limites de tamanho para dados em cache (alguns navegadores têm um limite de 5 MB por site).

**SUPORTE:** Internet Explorer 10, Firefox, Chrome, Safari e Opera suportam Application Cache.

## Questão 32

### FCC 2014 - AL-PE - Analista Legislativo - Sistemas

A linguagem HTML5 permite criar uma versão offline do site, através da criação de um arquivo de manifesto que indica quais os arquivos do site serão carregados em cachê para navegação mesmo se o usuário não estiver conectado à internet. Com relação ao uso dos recursos que permite isto, considere um site composto pelos arquivos index.html, config.css, banner.gif, rotina.js, page.appcache e acesso.jsp. Dois destes arquivos são apresentados a seguir:

Nos arquivos apresentados, as lacunas I e II são correta e, respectivamente, preenchidas por:

- a < html manifest="page.appcache" > e CACHE ON.
- b < html src="page.appcache" > e SESSION CACHE.
- c < html src="page.appcache" > e MANIFEST.
- d < html resource="page.appcache" > e DOWNLOAD.
- e < html manifest="page.appcache" > e CACHE MANIFEST.

index.html

<!DOCTYPE HTML>

I  
.....

<head><title>Home</title></home>

<body>

<!-- corpo da página aqui -->

</body>

</html>

page.appcache

II  
.....

/config.css

/banner.gif

/rotina.js

NETWORK:

acesso.jsp

FALLBACK:

/html/ /erro.html

## Questão 32

### FCC 2014 - AL-PE - Analista Legislativo - Sistemas

A linguagem HTML5 permite criar uma versão offline do site, através da criação de um arquivo de manifesto que indica quais os arquivos do site serão carregados em cachê para navegação mesmo se o usuário não estiver conectado à internet. Com relação ao uso dos recursos que permite isto, considere um site composto pelos arquivos index.html, config.css, banner.gif, rotina.js, page.appcache e acesso.jsp. Dois destes arquivos são apresentados a seguir:

Nos arquivos apresentados, as lacunas I e II são correta e, respectivamente, preenchidas por:

a < html manifest="page.appcache" > e CACHE ON.

b < html src="page.appcache" > e SESSION CACHE.

c < html src="page.appcache" > e MANIFEST.

d < html resource="page.appcache" > e DOWNLOAD.

➡ e < html manifest="page.appcache" > e CACHE MANIFEST.

index.html

<!DOCTYPE HTML>

I  
.....

<head><title>Home</title></home>

<body>

<!-- corpo da página aqui -->

</body>

</html>

page.appcache

II  
.....

/config.css

/banner.gif

/rotina.js

NETWORK:

acesso.jsp

FALLBACK:

/html/ /erro.html

## Questão 33

### FCC 2015 – CNMP - Analista de Tecnologia da Informação e Comunicação

Em um site desenvolvido com HTML5 existe um arquivo chamado teste.appcache, com a lista de arquivos que serão armazenados em cache para navegação off-line. O conteúdo deste arquivo é mostrado abaixo.

CACHE MANIFEST

index.html

stylesheet.css

logo.png

main.js

Para especificar o caminho deste arquivo em uma página HTML e consequentemente ativar o cache, deve ser utilizada a instrução

- a) < cache src="teste.appcache" offline="on">
- b) < html manifest="teste.appcache">
- c) < manifest src="teste.appcache" offline="on">
- d) < cache manifest="teste.appcache">
- e) < manifest url="teste.appcache" offline="on">



## Questão 33

### FCC 2015 – CNMP - Analista de Tecnologia da Informação e Comunicação

Em um site desenvolvido com HTML5 existe um arquivo chamado teste.appcache, com a lista de arquivos que serão armazenados em cache para navegação off-line. O conteúdo deste arquivo é mostrado abaixo.

CACHE MANIFEST

index.html

stylesheet.css

logo.png

main.js

Para especificar o caminho deste arquivo em uma página HTML e consequentemente ativar o cache, deve ser utilizada a instrução

a) < cache src="teste.appcache" offline="on">

➔ b) < html manifest="teste.appcache">

c) < manifest src="teste.appcache" offline="on">

d) < cache manifest="teste.appcache">

e) < manifest url="teste.appcache" offline="on">

# Gabarito

- |  |                                      |
|--|--------------------------------------|
| <b>1 - FCC 2014 AL-PE: D</b>           | <b>20 – CESPE 2014 ANTAQ - CERTO</b> |
| <b>2 – VUNESP 2014 PRODEST-ES: E</b>   | <b>21 – FCC 2011 TRE-PE - E</b>      |
| <b>3 – CESPE 2013 SERPRO: CERTO</b>    | <b>22 – FCC 2012 MPE-AP - C</b>      |
| <b>4 - FGV2015 - TJ-SC: B</b>          | <b>23 – FCC 2012 MPE-PE - C</b>      |
| <b>5 – CESPE 2014 – ANTAQ – ERRADO</b> | <b>24 – FCC 2014 TJ-AP - B</b>       |
| <b>6 – FCC 2014 TRT 1 R - C</b>        | <b>25 - CESPE2013 BACEN - CERTO</b>  |
| <b>7 – CESGRANRIO 2014 – EPE - A</b>   | <b>26 - CESPE2014 TJ-SE - CERTO</b>  |
| <b>8 – FCC 2014 – SABESP - B</b>       | <b>27 - FCC 2015 CNMP - E</b>        |
| <b>9 – FEPESE 2014 MPE-SC - C</b>      | <b>28 – CESPE 2013 MPOG - CERTO</b>  |
| <b>10 – FCC 2014 TRF 4 R - A</b>       | <b>29 – FJG 2014 Câmara RJ - B</b>   |
| <b>11 – CESPE 2014 ANTAQ - CERTO</b>   | <b>30 – FCC 2014 TRF 4 R - B</b>     |
| <b>12 – FCC 2012 TRF 2 R - D</b>       | <b>31 – FCC 2012 TRE-SP - C</b>      |
| <b>13 – CESPE 2012 TJ-RO - C</b>       | <b>32 – FCC 2014 AL-PE - E</b>       |
| <b>14 - CS UFG 2014 UEAP - C</b>       | <b>33 – FCC 2015 CNMP - B</b>        |
| <b>15 - CESPE 2013 TRE-MS - A</b>      |                                      |
| <b>16 - FMP-RS 2013 MPE-AC - A</b>     |                                      |
| <b>17 - CS UFG 2014 UEAP - D</b>       |                                      |
| <b>18 – CESPE 2013 STF - CERTO</b>     |                                      |
| <b>19 – FUNDEP 2014 IFN-MG - D</b>     |                                      |

# Fim

Para possíveis dúvidas sobre os assuntos desta revisão, questões de provas ou ajuda na elaboração de recursos, foi criada a lista de discussão abaixo:

[professor\\_vreis@googlegroups.com](mailto:professor_vreis@googlegroups.com)

Para elogios, sugestões de melhoria ou críticas, utilizar o e-mail encontrado no rodapé dos slides.