

HTML



Visão Geral

Preparativos - software:

O Editor favorito - sugestões:

Notepad++, VS Code, Sublime Text, NetBeans, JSFiddle (online), Codepen (online), W3Schools (online)

O Navegadores favorito. Ex:

Firefox 4 <http://www.mozilla.com/>

Chrome <http://www.google.com/chrome>

Opera <http://www.opera.com/>

Safari <http://www.apple.com/safari/>

MS Edge <https://www.microsoft.com/pt-br/edge>

Uma visão geral sobre

Nova estrutura, Forms, SVG, Canvas, WAI-ARIA, vídeo e áudio, armazenamento offline, drag & drop, mensagens, sockets e geolocalização

Porque HTML5 & CSS3?

Microsoft, Apple, Mozilla, Google, Facebook, YouTube, Vimeo, etc. – todos comprometidos com o HTML5 e o futuro da Web.

HTML5 representa uma Web mais **prática**, **semântica** e mais **funcional**.

CSS3 torna elementos visuais comuns fáceis, programáticos e não baseados em imagem

Suporte ao HTML5+CSS3 hoje:

Safari, Firefox, Chrome, Opera, IE9

Existem estratégias para browsers antigos.

A longa jornada



História do HTML

Pra quê? *para entender porque certas decisões foram tomadas e porque algumas coisas são como são.*

W3C & HTML até a v4.01

- Gerra dos Browser – disputa por características

- Standards process at the W3C – debate, committees, specifications

- “Shipping code wins”

Razões por trás do XHTML

XML foi visto como futuro da Web

XHTML surgiu com a função de **transição** no caminho do XML

Por que XHTML foi condenado?

Falta de compatibilidade anterior

Interpretação estranha do XML

Adesão estrita aos MIME types

text/html vs. aplicação/xhtml+xml

Todos navegadores permitem text/html

Problema no IE com aplicação/xhtml+xml

Princípios do HTML5

Compatibilidade com versões anteriores

Tratamento de erros bem definido

Aplicações e uso prático

Adoção da interação com JavaScript com a
marcação

Processo aberto

2 fontes para o HTML5

W3C spec: <http://www.w3.org/TR/html5/>

W3C Editor's Draft:

<http://dev.w3.org/html5/spec/Overview.html>



Iniciando com o HTML5!

Doctype - forma antiga:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD  
XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xh  
tml1-transitional.dtd">
```

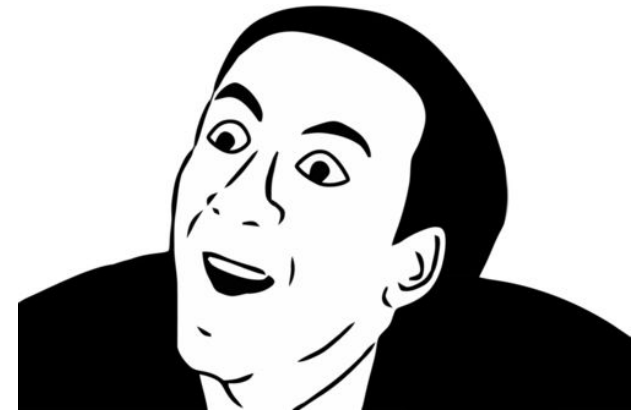
o !

O doctype é a maneira de você dizer para qualquer navegador como ele deve agir ao ler seu código HTML
É a primeira linha de código na página HTML



Doctype – em HTML5:

<!DOCTYPE html>



Definindo o **character set**

Por quê? Processamento do texto + preocupação com segurança:

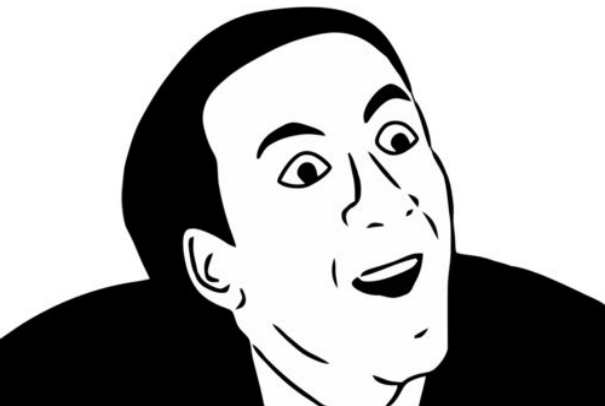
<http://code.google.com/p/doctype/wiki/ArticleUtf7>

Antigamente: Latin 1: `<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">`, ou qualquer charset regional necessário para o idioma (i.e. Big-5 para Chinês tradicional).

Unicode torna ajustes individuais de caracteres desnecessários: `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`

Charset no HTML5

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



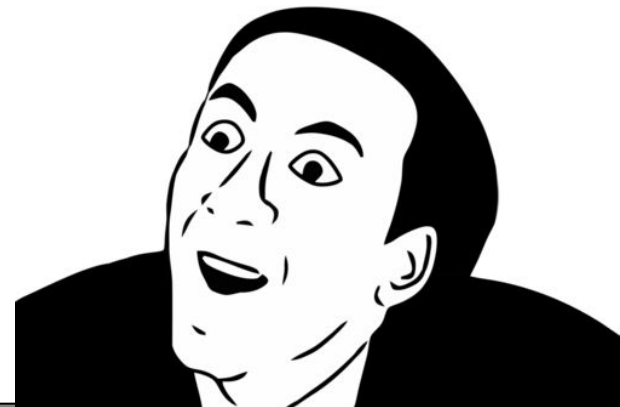
(Nota: `/>` tag de fechamento padrão necessário apenas para XHTML)

O elemento HTML, velha forma XHTML

```
<html  
xmlns="http://www.w3.org/1999/xhtml"  
lang="en" xml:lang="en">
```


O elemento HTML, estilo HTML5

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



usados pelos mecanismos de busca, sintetizadores de voz e navegadores para correção de hifenização regional e ortografia. Lista de códigos: [http://msdn.microsoft.com/en-us/library/ms533052\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms533052(VS.85).aspx)

HTML5 - Estrutura básica

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <title>Minha página</title>
    <meta charset="utf-8">
  </head>
  <body>
    <p>Esta é a minha página.</p>
  </body>
</html>
```

Aparece na guia da janela

Cabeçalho do documento
Meta tags, partes que
não aparecem
(muito importante)

Interpreta os caracteres latinos, ~, ', ^, ç, etc..

Corpo da página
Onde vai tudo o que
Aparece (textos, tabelas,
Imagens, listas, forms...)

Os novos elementos HTML5

datalist

keygen

output

ruby, rt e rp

wbr

embed

command

details

article

aside

figure

footer

header

hgroup

nav

section

mark

meter

progress

time

canvas

video

Novos atributos HTML5

Alguns deles:

placeholder – máscara no formulário (muito útil)

autofocus – atribui foco aos campos de texto

contenteditable – torna um dado elemento editável. Experimentar ;-).

reversed – funciona em `` invertendo a ordem.

Elementos HTML5 modificados

<a> agora pode envolver em torno de elementos de bloco

**** e **<i>** voltaram.

<cite> definido para ser título de um trabalho

<hr> agora é uma "quebra temática em nível de parágrafo"

<small> textos pequenos, "copyright, créditos..."

Elementos removidos

basefont

big

center

font

strike

tt

u

frame

frameset

noframes

acronym (abbr)

applet (object)

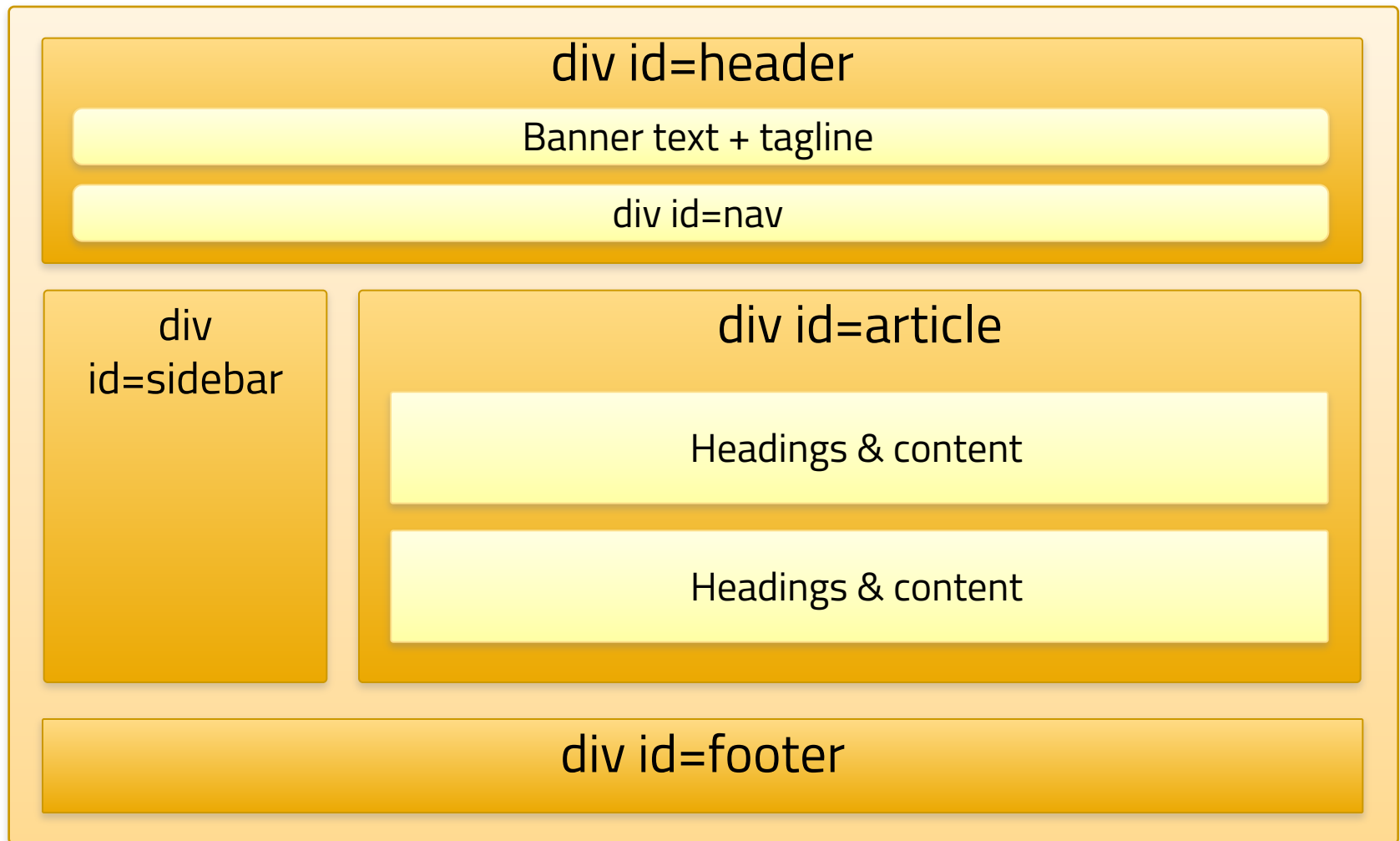
Estrutura típica da Página HTML 4

```
<div id="header">
  <h1>Tales of Woe</h1>
  <p class="tagline">The neverending saga continues...</p>
  <div id="nav">
    <h2>Navigation</h2>
    <ul>
      <li><a href="one.html">Page One</a></li>
      <li><a href="two.html">Page Two</a></li>
      <li><a href="three.html">Page the Third</a></li>
    </ul>
  </div>
</div>

<div class="article">
  <h2>The Stagnant Staircase</h2>
  <p>It was a dark and stormy night...</p>
</div>

<div id="footer">
  <p>©2010 In Your Nightmares Productions.</p>
</div>
```

Como aparece a estrutura HTML 4



Estrutura típica HTML5

<header>

<hgroup>

`<h1>Tales of Woe</h1>`

`<h2>The neverending saga continues... </h2>`

</hgroup>

<nav>

`<h2>Navigation</h2>`

``

`Page One`

`Page Two`

`Page the Third`

``

`</nav>`

`</header>`

<article>

`<h2>The Stagnant Staircase</h2>`

`<p>It was a dark and stormy night... </p>`

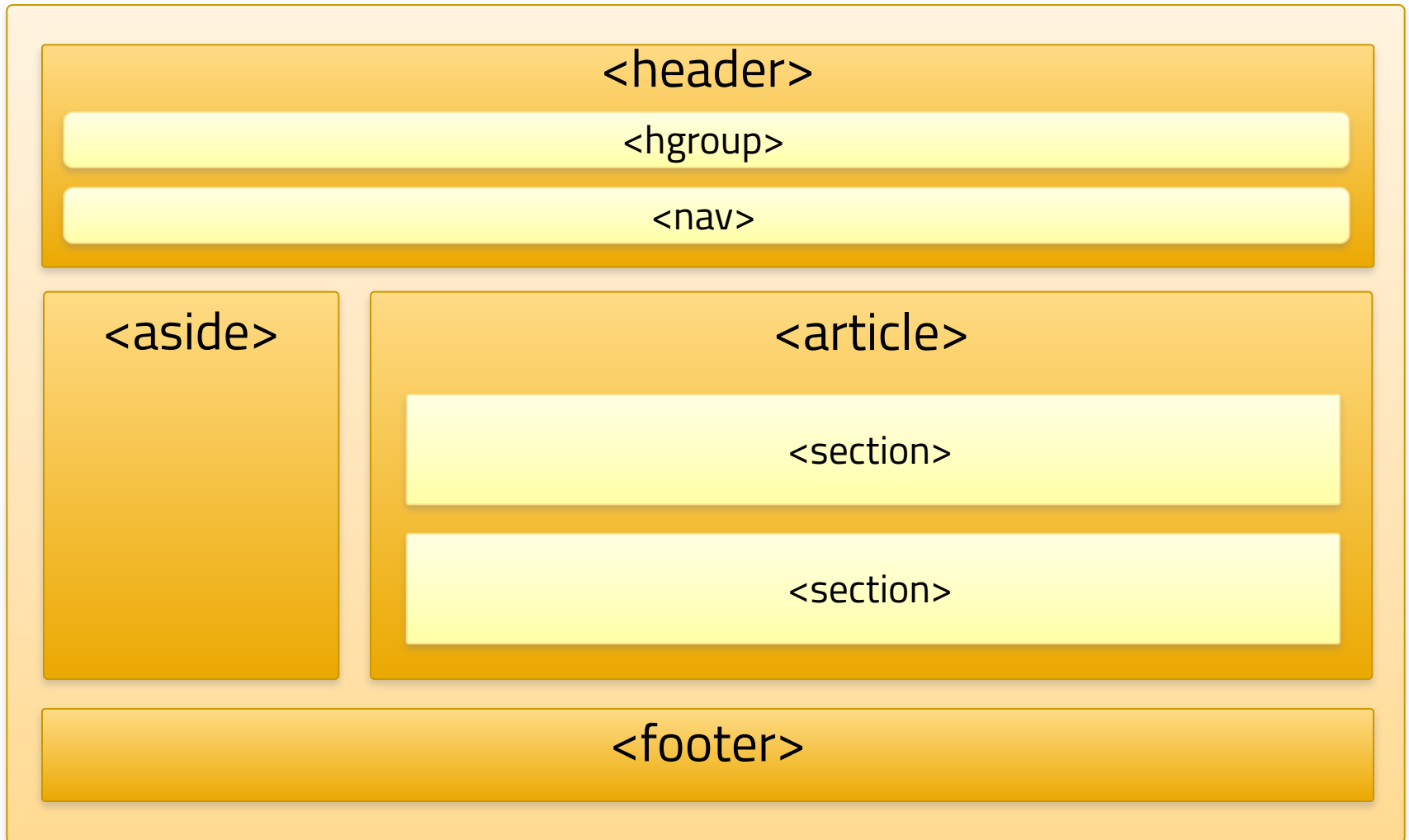
`</article>`

<footer>

`<p>©2010 In Your Nightmares Productions. </p>`

`</footer>`

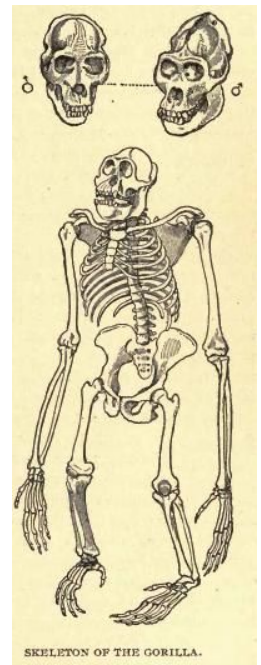
Elementos estruturais em HTML5



Seccionando títulos (código)

```
<section>
  <h1>Homininae</h1>
  <section>
    <h1>Gorillini</h1>
    <section>
      <h1>Gorilla gorilla</h1>
      <section>
        <h1>Gorilla gorilla gorilla</h1>
        <h1>Gorilla gorilla diehli</h1>
      </section>
      <h1>Gorilla beringei</h1>
      <section>
        <h1>Gorilla beringei beringei</h1>
        <h1>Gorilla beringei graueri</h1>
      </section>
    </section>
  </section>
  <section>
    <h1>Hominini</h1>
    <section>
      <h1>Pan</h1>
    </section>
  </section>
```

```
<h1>Pan troglodytes</h1>
<section>
  <h1>Pan troglodytes troglodytes</h1>
  <h1>Pan troglodytes verus</h1>
  <h1>Pan troglodytes vellerosus</h1>
  <h1>Pan troglodytes schweinfurthii</h1>
</section>
<h1>Pan paniscus</h1>
</section>
</section>
<section>
  <h1>Homo</h1>
  <section>
    <h1>Homo sapiens</h1>
    <section>
      <h1>Homo sapiens sapiens</h1>
    </section>
  </section>
</section>
</section>
```



Seccionando títulos (resultado)

Homininae

Gorillini

Gorilla gorilla

Gorilla gorilla gorilla

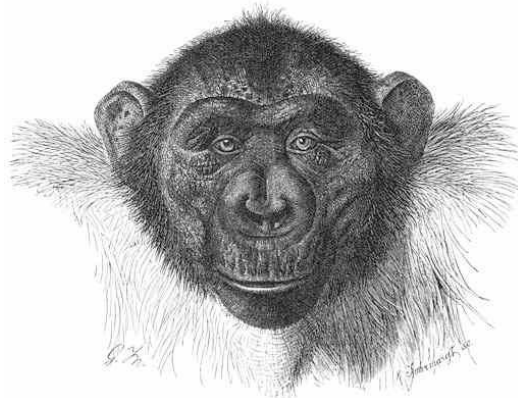
Gorilla gorilla diehli

Gorilla beringei

Gorilla beringei beringei

Gorilla beringei graueri

Hominini



Pan

Pan troglodytes

Pan troglodytes troglodytes

Pan troglodytes verus

Pan troglodytes vellerosus

Pan troglodytes

schweinfurthii

Pan paniscus

Homo

Homo sapiens

Homo sapiens sapiens

Elemento figure

```
<figure>
```

```
  
```

```
  <figcaption>Uma Viola Amati  
1710 viola.</figcaption>
```

```
</figure>
```

<aside> de Prata

```
<aside>
```

```
  <h2>Definições</h2>
```

```
  <dl>
```

```
    <dt>Prata</dt>
```

```
    <dd>Metal precioso</dd>
```

```
    <dd>Gíria para dinheiro</dd>
```

```
  </dl>
```

```
</aside>
```

Dentro de um article: relacionado ao seu conteúdo (caixa de informação)

Fora de um article: conteúdo relevante da própria página (barra lateral - sidebar)

Aplicações XML no HTML5

SVG e MathML estão disponíveis no HTML5

Navegadores: Firefox 4 ok, WebKit geralmente funciona, outros browsers - em desenvolvimento

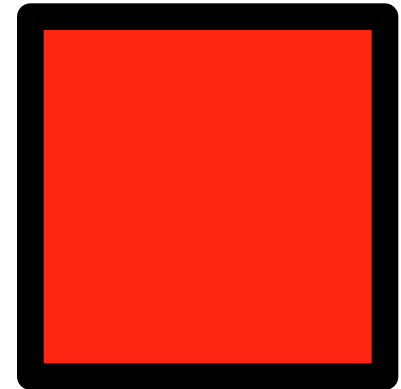
Namespaces não são necessários

imagem vetorial, não distorce com o zoom



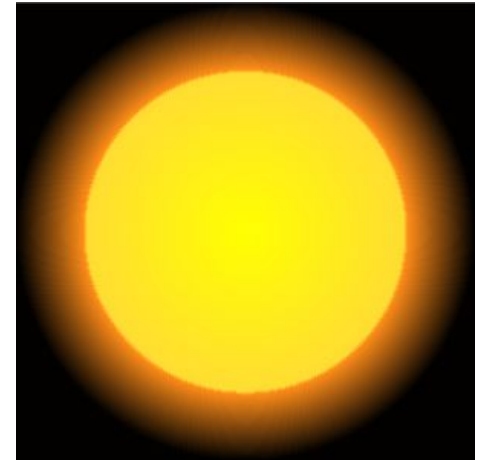
SVG - Exemplo 1

```
<svg width="300px" height="300px">  
  <rect x="10px" y="10px"  
    width="250px" height="250px"  
    fill="red" stroke="black"  
    stroke-width="10px" />  
</svg>
```



SVG - Exemplo 2

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>O Sol em SVG</title>
  </head>
  <body>
    <h1>O Sol em SVG</h1>
    <svg style="width:300px; height:300px;">
      <radialGradient cx="0.5" cy="0.5" r="0.5" id="g">
        <stop stop-color="rgb(255, 255, 0)" offset="0"/>
        <stop stop-color="rgb(255, 221, 51)" offset="0.7"/>
        <stop stop-color="rgb(254, 140, 25)" offset="0.7"/>
        <stop stop-color="rgb(0, 0, 0)" offset="1"/>
      </radialGradient>
      <rect width="100%" height="100%" fill="url(#g)"/>
    </svg>
  </body>
</html>
```



MathML Exemplo

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>The Dirac &delta;-
    function</title>
</head>
<body>
  <h1>The Dirac &delta;-
    function</h1>
  <math style="font-
    size:200%">
    <mrow>
      <msubsup>
        <mo>&int;</mo>
```

```
<mrow>
  <mrow>
    <mo>-</mo>
    <mi>&infin;</mi>
  </mrow>
</mrow>
<mi>&infin;</mi>
</msubsup>
<mrow>
  <mi>f</mi>
  <mi>&delta;</mi>
</mrow>
</mrow>
<mo>=</mo>
<mrow>
```

```
<mo>f</mo>
<mfenced>
  <mrow>
    <mn>0</mn>
  </mrow>
</mfenced>
</mrow>
</math>
</body>
</html>
```

$$\int_{-\infty}^{\infty} f \delta = f(0)$$

O elemento Canvas

Renderização de imagens via script

Comparando com SVG:

- Melhor performance no Canvas.

- SVG é XML, enquanto Canvas é principalmente JS.

- SVG tem acesso ao DOM.

- Canvas é mais direcionado a bitmap. SVG, vetor

Canvas básico

```
<canvas id="example" width="250"  
height="250">
```

Texto descritivo

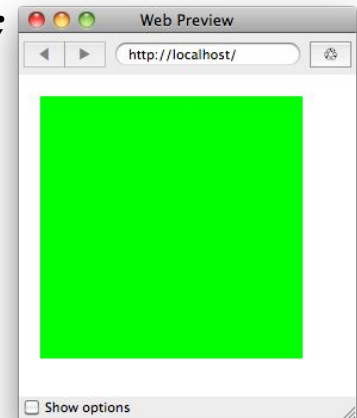
```
</canvas>
```

Possível CSS em vez dos atributos width e height

"Texto descritivo" usado para agentes que não interpretam <canvas>

Trecho de `<canvas>` verde

```
<script>
  function draw() {
    var canvas = document.getElementById("canvas");
    if (canvas.getContext) {
      var ctx = canvas.getContext("2d");
      ctx.fillStyle = "rgb(0,255,0)";
      ctx.fillRect (10, 10, 220, 220);
    }
  }
</script>
<body onload="draw();" >
  <canvas id="canvas" width="250" height="250">
    Isto deve ser um quadrado verde.</canvas>
</body>
```



ARIA no HTML5

ARIA: Accessible Rich Internet Applications

API que possibilita tecnologias assistivas acessar controles de aplicativos (ex. Essa imagem é um checkbox)

Alvo HTML, funcionalidade do navegador, JavaScript/Ajax

Inserir entradas para acessibilidade hooks em atributos HTML.

Usa o atributo **aria-** como prefixo

Suportado no IE, Firefox, Opera, Chrome, Safari, principais fornecedores de leitor de tela, Dojo, YUI, jQuery

ARIA no HTML5 - exemplo

```
<ul id="tree1"
    role="tree"
    tabindex="0"
    aria-
    labelledby="label_1">
  <li role="treeitem"
      tabindex="-1"
      aria-expanded="true">
    Hominids</li>
  <li role="group">
    <ul>
      <li role="treeitem"
          tabindex="-1">
            Neandertal</li>
      <li role="treeitem"
          tabindex="-1">
            Homo Sapiens</li>
    </ul>
  </li>
</ul>
```

Recursos ARIA

Bruce Lawson's Redesigning with HTML5 and ARIA

<http://www.brucelawson.co.uk/2009/redesigning-with-html-5-wai-aria/>

WAI-ARIA Overview:

<http://www.w3.org/WAI/intro/aria.php>

WAI-ARIA Primer:

<http://www.w3.org/TR/wai-aria-primer/>

ARIA basics at Standardista.com:

<http://www.standardista.com/standards/wai-aria-accessible-rich-internet-applications-basics>

HTML5 novidades nos formulários

Novos atributos

Required

Autofocus

Placeholder

Form

Novos Input types

E-mail

Date

Range

Search

Tel

Color

- Number

- Novos elementos

- `<meter>`

- `<progress>`

- `<output>`

Novos atributos <input>

min

max

multiple

pattern

step

autocomplete

autofocus

required

O atributo **required**

Usado nos elementos:

```
<input type="text" name="moof" required="required" />
```

Sendo mais conciso:

```
<input type=text name=moof required>
```

O atributo **aria-required** sinaliza entrada para dispositivos assistivos.

Seletores CSS:

```
[required] { background-color: whitesmoke; }  
:invalid { background-color: #F66; }
```

Saindo do <form>

Usar atributo **form** para ligar elementos ao formulário:

```
<form id="perfil">
```

Insira seu <input type="email" name="email" placeholder="Endereço de E-mail">.

```
</form>
```

```
<textarea form="perfil">
```

Escreva aqui sobre você.

```
</textarea>
```

Baseado no material de

Joseph R. Lewis

Chief Web Architect, Sandia National
Laboratories

Web development for 13 years

Author, AdvancED CSS and Foundation
Website Creation

