

HTML5 & CSS3

Jonas Keizo Hirata



O QUE VEREMOS?

- ▶ Uma breve história do HTML
- ▶ Estrutura de um documento HTML5
- ▶ Comparação entre HTML4/XHTML e HTML5
- ▶ Apresentar a especificação
- ▶ Descobrir as novas possibilidades

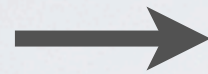
O QUE NÃO VEREMOS?

- ▶ Código e mais código
- ▶ Explicação detalhada de uso das APIs
- ▶ Exemplos de Elementos e ou APIs que não foram implementados por nenhum browser
- ▶ Explicações passo a passo de técnicas de como tentar tornar um browser compatível com o HTML5

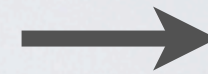
1998: HTML 4.01

2000: XHTML

Transitional
Strict
~~Frameset~~



Geração de desenvolvedores preocupados com código válido e bem estruturado.

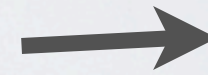


2002: XHTML 2.0

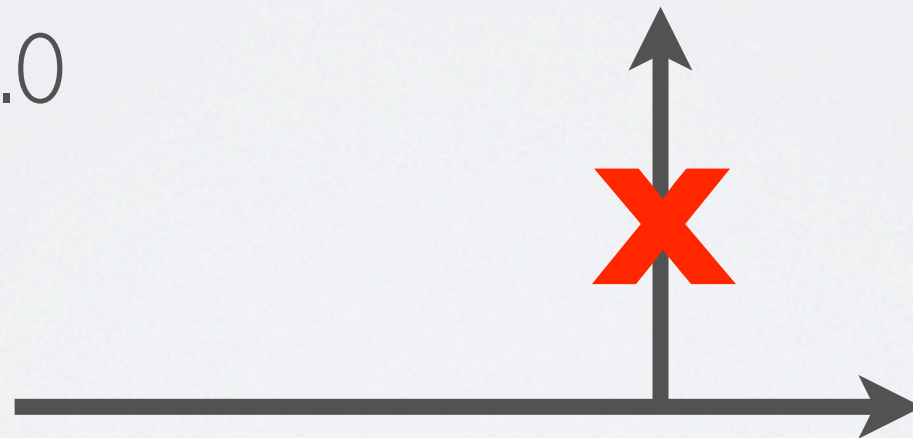
Prometia uma revolução na linguagem



Quebraria a compatibilidade com browsers mais antigos



2004: Web Forms 2.0



WHATWG
Web Hypertext Application
Technology Working Group

www.whatwg.org

2006: W3C cai na real (~~XHTML 2.0~~)



Processo de especificação totalmente diferente: aberto e empírico

2009: W3C direciona os esforços para o HTML5

2010 ~ 2011: Suporte ao HTML5 aumenta a cada release dos browsers.

2012: O mundo acaba :P

Vamos começar pelo começo e por aquilo que ninguém lembra de cabeça como se escreve: **DOCTYPE**

HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
```

```
"http://www.w3.org/TR/html4/strict.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
```

```
"http://www.w3.org/TR/html4/frameset.dtd">
```


Vamos começar pelo começo e por aquilo que ninguém lembra de cabeça como se escreve: **DOCTYPE**

HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

XHTML 1.0

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

Vamos começar pelo começo e por aquilo que ninguém lembra de cabeça como se escreve: **DOCTYPE**

HTML5

```
<!doctype html>
```


Outro elemento importante em qualquer documento HTML (HTML 4.01/XHTML/HTML5): **META**

HTML 4.01

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
```

Outro elemento importante em qualquer documento HTML (HTML 4.01/XHTML/HTML5): **META**

HTML 4.01

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
```

XHTML

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
```


Outro elemento importante em qualquer documento HTML (HTML 4.01/XHTML/HTML5): **META**

HTML 4.01

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
```

XHTML

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
```

HTML5

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

Hora de juntar tudo...

```
<!doctype html>  
<html lang="pt-BR">  
  <head>  
    <meta charset="utf-8">  
    <title>Título da página</title>  
  </head>  
  <body>  
    <!-- Conteúdo aqui -->  
  </body>  
</html>
```


E aquele negócio de SGML e XML?

E aquele negócio de SGML e XML?

HEIN?!!

E aquele negócio de SGML e XML?

```
<!doctype html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8">
    <title>Título</title>
  </head>
  <body>
    <p>Texto simples.
  </body>
</html>
```

```
<html xmlns="http://www.w3.org/
1999/xhtml" xml:lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <title>Título</title>
  </head>
  <body>
    <p>Texto simples.</p>
  </body>
</html>
```

E aquele negócio de SGML e XML?

```
<!doctype html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8">
    <title>Título</title>
  </head>
  <body>
    <p>Texto simples.
  </body>
</html>
```

text/html



HTML

Parser/Serializer



Página

```
<html xmlns="http://www.w3.org/
1999/xhtml" xml:lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <title>Título</title>
  </head>
  <body>
    <p>Texto simples.</p>
  </body>
</html>
```

application/xhtml+xml



XML

Parser/Serializer



Como um blog se parece?

Como um blog se parece? Um possível wireframe seria...

Cabeçalho (header)

Post

Menu
lateral
(sidebar)

Post

Rodapé (footer)

E em HTML, como ficaria?

E em HTML, como ficaria? Provavelmente assim...

```
<div id="header">
  <h1>Meu BLOG</h1>
</div>
<div id="sidebar">
  <ul>
    <li><a href="maio-2011.html">Maio 2011</a></li>
    <li><a href="abril-2011.html">Abril 2011</a></li>
    <li><a href="marco-2011.html">Marco 2011</a></li>
    <li><a href="fevereiro-2011.html">Fevereiro 2011</a></li>
    <li><a href="janeiro-2011.html">Janeiro 2011</a></li>
  </ul>
</div>
<div class="post">
  <h2>Título do Post 1</h2>
  <p>Texto do Post 1</p>
</div>
<div class="post">
  <h2>Título do Post 2</h2>
  <p>Texto do Post 2</p>
</div>
<div id="footer">
  <p><small>Meu BLOG 2011. Nenhum direito reservado.</small></p>
</div>
```


Apenas para visualizarmos o HTML em nosso wireframe...

```
<div id="header">
```

```
<div class="post">
```

```
<div id="sidebar">
```

```
<div class="post">
```

```
<div id="footer">
```

Que diabos é esse tal
<div>?!

De acordo com a especificação do HTML 4.01:

“The DIV and SPAN elements, in conjunction with the id and class attributes, offer a generic mechanism for adding structure to documents. These elements define content to be inline (SPAN) or block-level (DIV) but impose no other presentational idioms on the content. Thus, authors may use these elements in conjunction with style sheets, the lang attribute, etc., to tailor HTML to their own needs and tastes.”

De acordo com a especificação do HTML 4.01:

“The DIV and SPAN elements, in conjunction with the id and class attributes, offer a generic mechanism for adding structure to documents. These elements define content to be inline (SPAN) or block-level (DIV) but impose no other presentational idioms on the content. Thus, authors may use these elements in conjunction with style sheets, the lang attribute, etc., to tailor HTML to their own needs and tastes.”

HEIN?! Inline?! Block-level?! Calma...

Uma palavrinha sobre semântica HTML...

A semântica HTML é o uso da linguagem de marcação HTML para dar significado ao conteúdo de um documento ao invés de definir a sua apresentação (aparência).

Uma palavrinha sobre semântica HTML...

A semântica HTML é o uso da linguagem de marcação HTML para dar significado ao conteúdo de um documento ao invés de definir a sua apresentação (aparência).

Exemplo:

`<i>`: formata um texto em itálico → apresentação

``: indica que o texto deve ser enfatizado → significado

`<cite>`: indica uma citação → significado

OK, mas e aquele tal
<div>?!

Apenas para visualizarmos o HTML em nosso wireframe...

```
<div id="header">
```

```
<div class="post">
```

```
<div class="post">
```

```
<div id="sidebar">
```

```
<div id="footer">
```


O elemento `<div>` **não possui significado** e, por padrão, **não afeta a apresentação.**

Mais uma olhadinha na especificação do HTML 4.01:

*“The DIV and SPAN elements, in conjunction with the id and class attributes, **offer a generic mechanism for adding structure to documents.** These elements define content to be inline (SPAN) or block-level (DIV) but impose no other presentational idioms on the content. Thus, authors may use these elements in conjunction with style sheets, the lang attribute, etc., to tailor HTML to their own needs and tastes.”*

Voltando ao nosso blog...

Não seria perfeito se fosse assim?

<header>

<post>

<sidebar>

<post>

<footer>

Por quê?

Por quê?

Dois motivos bem simples:

- 1 - obviamente, escreve-se menos
- 2 - traria significado ao conteúdo

Legal... e daí?

Legal... e daí? É aí que entra o HTML5!

Um dos objetivos do HTML5 é trazer uma semântica maior aos elementos da linguagem. Para isso foram introduzidos novos elementos e alguns já existentes foram alterados semanticamente.

Vamos ver como poderia ficar o nosso blog em HTML5:

```
<header>
  <h1>Meu BLOG</h1>
</header>
<aside>
  <nav>
    <ul>
      <li><a href="maio-2011.html">Maio 2011</a></li>
      <li><a href="abril-2011.html">Abril 2011</a></li>
      <li><a href="marco-2011.html">Marco 2011</a></li>
      <li><a href="fevereiro-2011.html">Fevereiro 2011</a></li>
      <li><a href="janeiro-2011.html">Janeiro 2011</a></li>
    </ul>
  </nav>
</aside>
<article>
  <h2>Título do Post 1</h2>
  <p>Texto do Post 1</p>
</article>
<article>
  <h2>Título do Post 2</h2>
  <p>Texto do Post 2</p>
</article>
<footer>
  <p><small>Meu BLOG 2011. Nenhum direito reservado.</small></p>
</footer>
```


Apenas para visualizarmos o HTML5 em nosso wireframe...

`<header>`

`<article>`

`<article>`

`<footer>`

`<aside>`

As tags `<post>` e `<sidebar>` que utilizamos na nossa “situação perfeita” não existem.

No lugar delas utilizamos as tags `<article>` e `<aside>` que são mais genéricas, afinal seria impossível existirem tags para para todo tipo de site com diversos temas diferentes, não é mesmo?

Analysis of Variance

$$BSS = \sum n_j \left(\bar{X}_j - M \right)^2 = \sum \left[\left(\sum X_j \right)^2 / n_j \right] - \left(\sum X_T \right)^2 / N$$

$$WSS = \sum \sum \left(X_{ij} - \bar{X}_j \right)^2 = \sum (n_j - 1) (s_j^2) = \sum X^2 - \sum \left[\left(\sum X_j \right)^2 / n_j \right]$$

$$TSS = (N - 1) (s_T^2) = \sum \sum (X_{ij} - M)^2 = \sum X_T^2 - \left(\sum X_T \right)^2 / N$$

$$F_{(df_b, df_w)} = \frac{BSS/k - 1}{WSS/n - k}$$

$$(df_b = k - 1, df_w = n - k, df_t = n - 1)$$

NOTE:

T = Total

M = Grand Mean

N = Overall N

TSS = BSS + WSS

Analysis of Variance

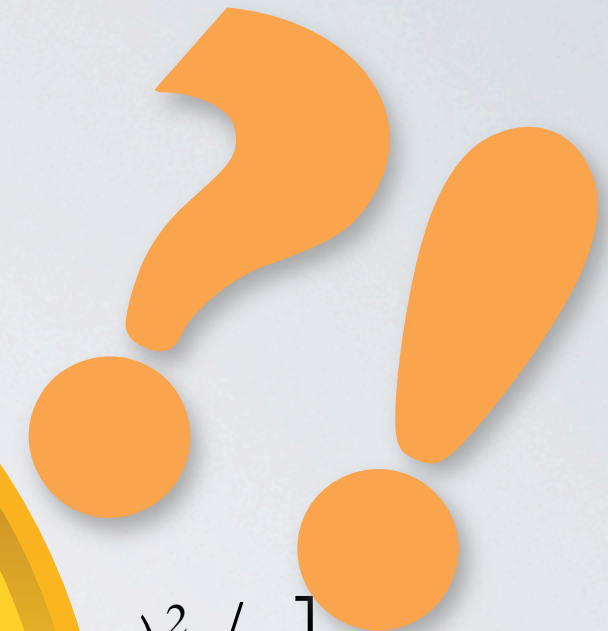
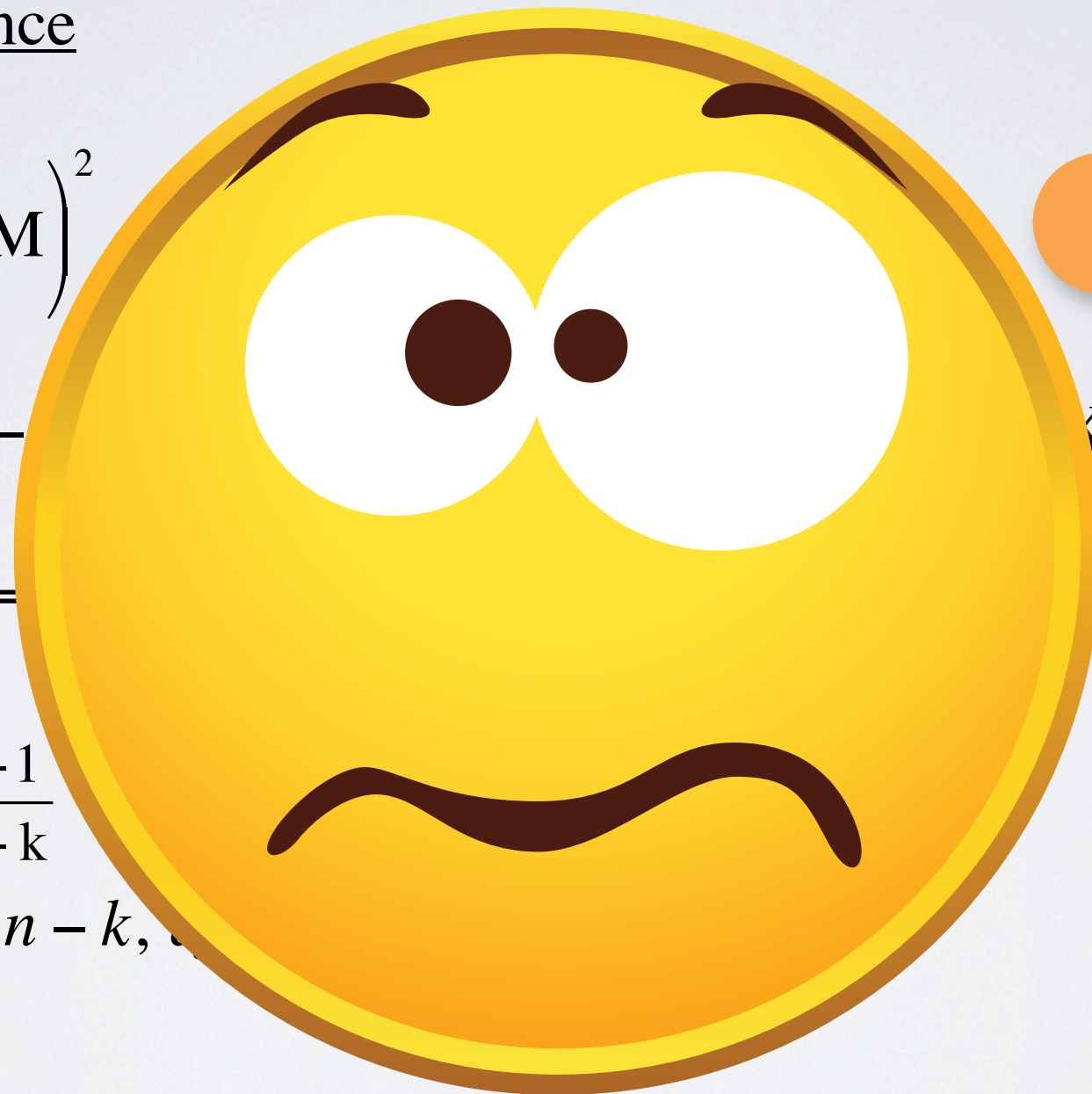
$$BSS = \sum n_j (\bar{X}_j - M)^2$$

$$WSS = \sum \sum \left[(X_{ij} - \bar{X}_j)^2 / n_j \right]$$

$$TSS = (N - 1)(s_T^2) =$$

$$F_{(df_b, df_w)} = \frac{BSS/k - 1}{WSS/n - k}$$

$$(df_b = k - 1, df_w = n - k, \dots)$$



T = Total
M = Grand Mean
N = Overall N
TSS = BSS + WSS

Analysis of Variance

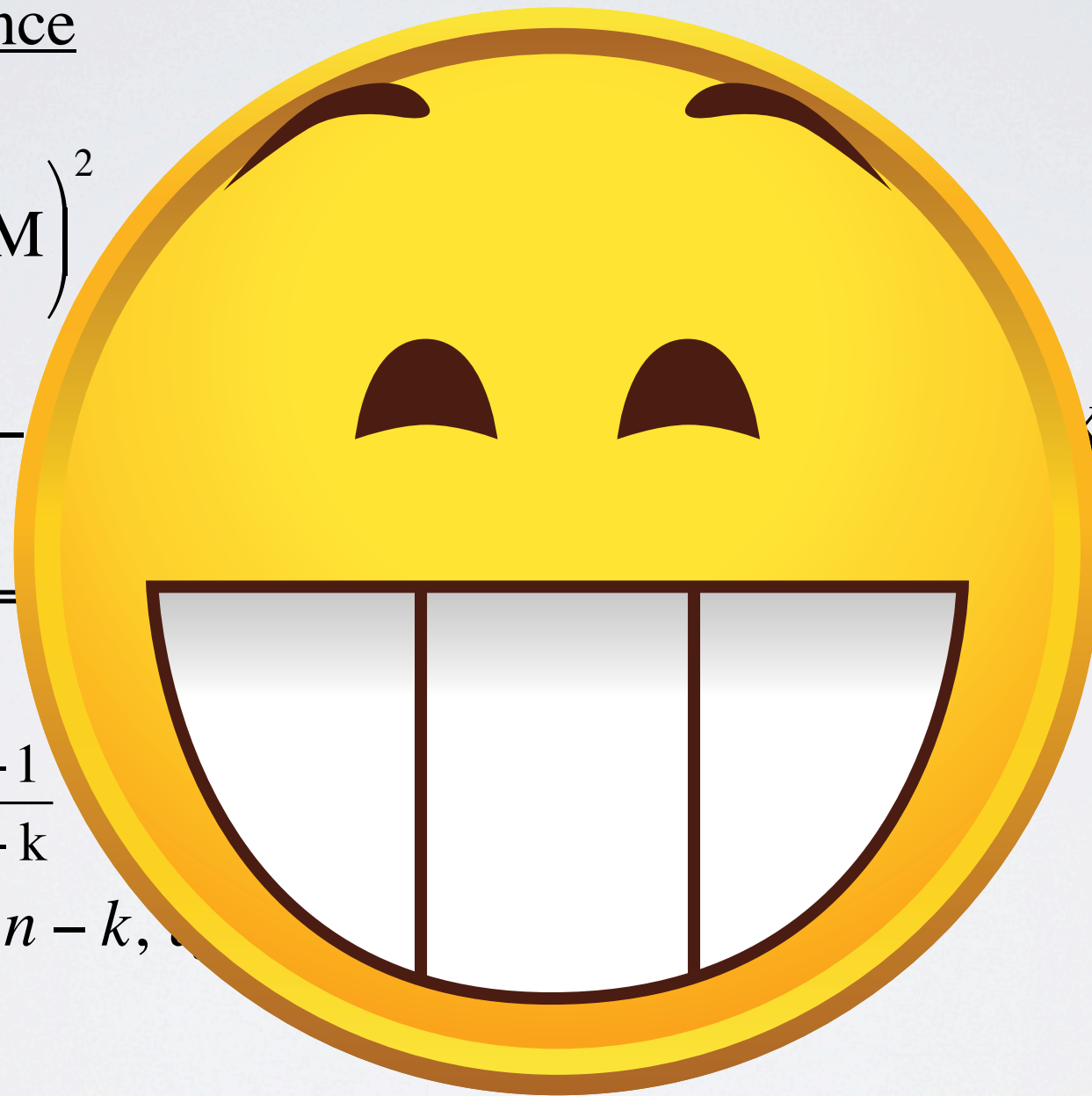
$$BSS = \sum n_j (\bar{X}_j - M)^2$$

$$WSS = \sum \sum \left[(X_{ij} - \bar{X}_j)^2 / n_j \right]$$

$$TSS = (N - 1)(s_T^2) =$$

$$F_{(df_b, df_w)} = \frac{BSS/k - 1}{WSS/n - k}$$

$$(df_b = k - 1, df_w = n - k, \dots)$$



T = Total
M = Grand Mean
N = Overall N
TSS = BSS + WSS

Valores populares para o atributo `id`

Total: 1.782.769

Valor	Frequência
<code>footer</code>	288.061
<code>content</code>	228.661
<code>header</code>	223.726
<code>logo</code>	121.351
<code>container</code>	119.877
<code>main</code>	106.327

Valor	Frequência
<code>table1</code>	101.677
<code>menu</code>	96.161
<code>layer1</code>	93.920
<code>autonumber1</code>	77.350
<code>search</code>	74.887
<code>nav</code>	72.057

Valor	Frequência
<code>wrapper</code>	66.730
<code>top</code>	66.615
<code>table2</code>	57.934
<code>layer2</code>	56.823
<code>sidebar</code>	52.416
<code>image1</code>	48.922

Fonte: Opera MAMA crawler

<http://dev.opera.com/articles/view/mama-common-attributes/>

MAMA - Metadata Analysis and Mining Application

Valores populares para o atributo `class`

Total: 2.139.184

Valor	Frequência
footer	179.528
menu	146.673
style1	138.308
msonormal	123.374
text	122.911
content	113.951
title	91.957

Valor	Frequência
style2	89.851
header	89.274
copyright	86.979
button	81.503
main	69.620
style3	69.349
small	68.995

Valor	Frequência
nav	68.634
clear	68.571
search	59.802
style4	56.032
logo	48.831
body	48.052
left	47.822

Fonte: Opera MAMA crawler

<http://dev.opera.com/articles/view/mama-common-attributes/>

MAMA - Metadata Analysis and Mining Application

Coincidência alguns desses nomes serem os mesmos que os utilizados em alguns dos novos elementos do HTML5?

Coincidência alguns desses nomes serem os mesmos que os utilizados em alguns dos novos elementos do HTML5?

Não

Lembra que eu disse que o processo de especificação do HTML5 era diferente? Eu disse que era:

Aberto

Qualquer um pode participar da lista de e-mail do WHATWG

Empírico

Os responsáveis pela especificação não levaram em conta apenas o que era discutido nas listas de e-mail ou o que eles achavam melhor. Eles observaram o comportamento dos desenvolvedores, ou seja, analisaram páginas e mais páginas em busca de um padrão.

De marcação:

<article>

<aside>

<figure>

<figcaption>

<footer>

<header>

<hgroup>

<mark>

<nav>

<progress>

<ruby>

<rt>

<rp>

<section>

<time>

<wbr>

De marcação:

<article>

<aside>

<figure>

<figcaption>

<footer>

<header>

<hgroup>

<mark>

<nav>

<progress>

<ruby> → **X** →



Ruby: Linguagem
de Programação

<rt>

<rp>

<section>

<time>

<wbr>

Mídia:

<audio>

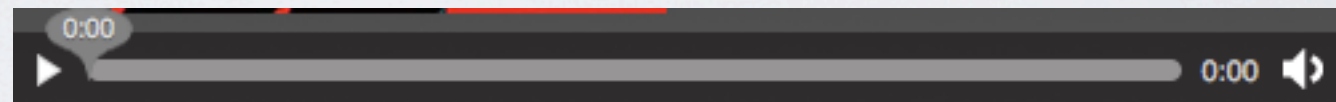
<video>

<source>

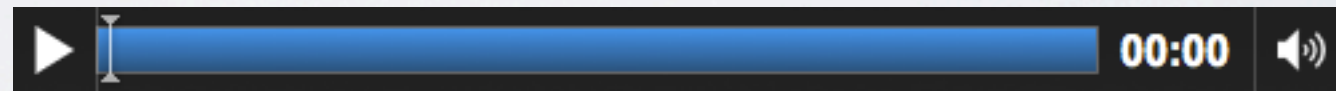
<embed>



Safari 5.0.5



Firefox 4.0.1



Chrome 11.0.696.68



Opera 11.11

Gráfico:

`<canvas>`

Gráfico:

<canvas>

```
// Get the canvas element.
var elem = document.getElementById('myCanvas');
if (!elem || !elem.getContext) {
    return;
}

// Get the canvas 2d context.
var context = elem.getContext('2d');
if (!context) {
    return;
}

context.fillStyle = '#00f';
context.strokeStyle = '#f00';
context.lineWidth = 4;

// Draw a right triangle.
context.beginPath();
context.moveTo(10, 10);
context.lineTo(100, 10);
context.lineTo(10, 100);
context.lineTo(10, 10);
context.fill();
context.stroke();
context.closePath();
```

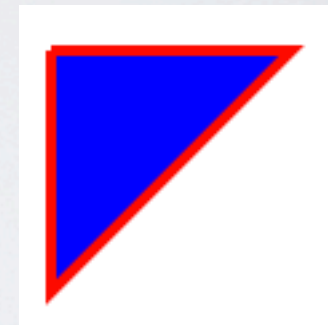
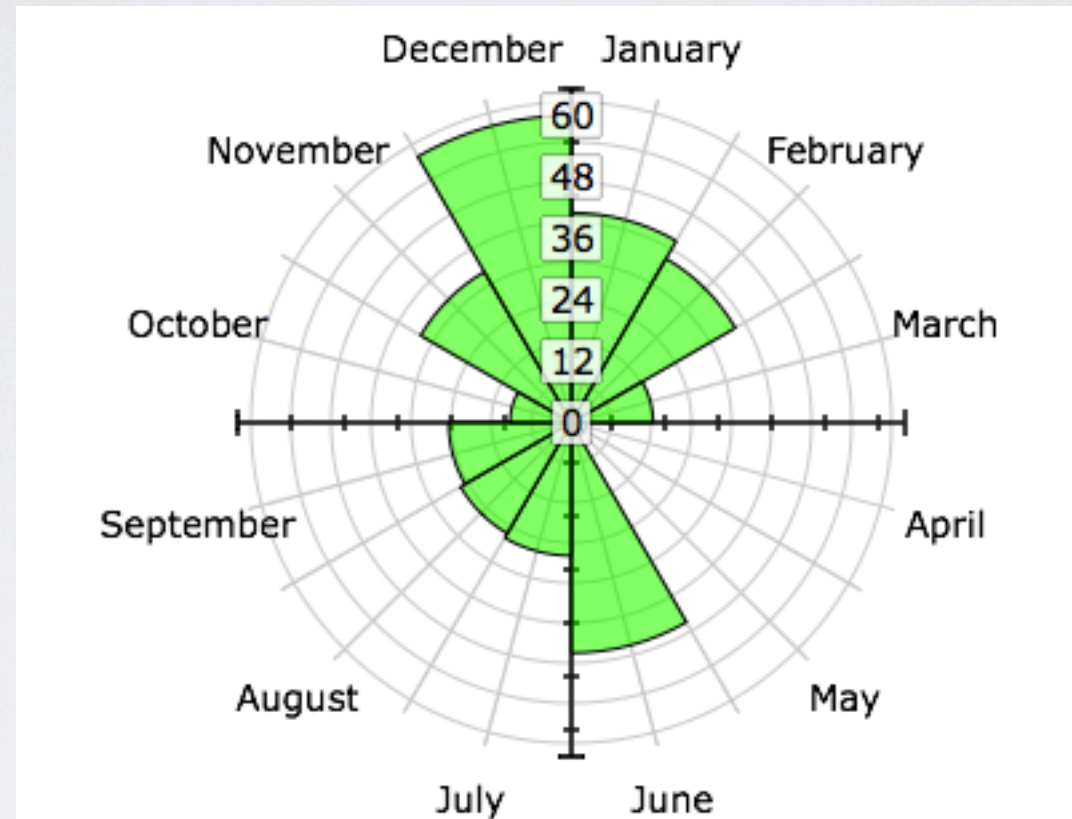


Gráfico:

<canvas>



Bibliotecas para manipular o canvas:

gury - <http://guryjs.org/>

EaselJS - <http://easeljs.com/>

RGraph - <http://www.rgraph.net/>

Processing.js - <http://processingjs.org/>

Formulário:

<datalist>

<keygen>

<output>

<dl>

<cite>

<address>

, <i>

,

<hr>

<small>

```
<input type="TIPO">
```

button

checkbox

color

date

datetime

datetime-local

email

file

hidden

image

month

number

password

radio

range

reset

search

submit

tel

text

time

url

week

<input>

Alguns dos novos atributos para o <input>:

list

autofocus

placeholder

required

multiple

pattern

autocomplete*

min

max

step

<input>

Alguns dos novos atributos para o <input>:

list

autofocus

placeholder

required

multiple

pattern

autocomplete*

min

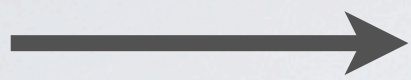
max

step

* The off state indicates either that the control's input data is particularly sensitive (for example the **activation code for a nuclear weapon**); or that it is a value that will never be reused (for example a one-time-key for a bank login) and the user will therefore have to explicitly enter the data each time, instead of being able to rely on the UA to prefill the value for him; or that the document provides its own autocomplete mechanism and does not want the user agent to provide autocompletion values.

contenteditable

data-* →



draggable="true"

spellcheck

tabindex="-1"

setAttribute()

getAttribute()

Hoje em dia, como armazenamos dados localmente através de um documento HTML?

Hoje em dia, como armazenamos dados localmente através de um documento HTML?

Uma dica...



s

Hoje em dia, como armazenamos dados localmente através de um documento HTML?

COOKIES



Compartilhado com o servidor através do cabeçalho de requisição



Chato de criar, chato de manter e chato de remover
Para armazenar estruturas de dados um pouco mais complexas, deve ser serializado primeiro

Quais as opções do HTML5?

▶ COOKIES

▶ Web Storage

▶ Web SQL Databases

E aí, já posso sair usando o HTML5?!

Sim

E aí, já posso sair usando o HTML5?!

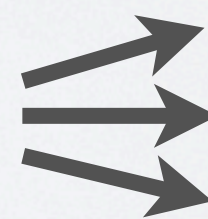
Sim... e Não

Utilize o bom senso

E aí, já posso sair usando o HTML5?!

Sim... e Não

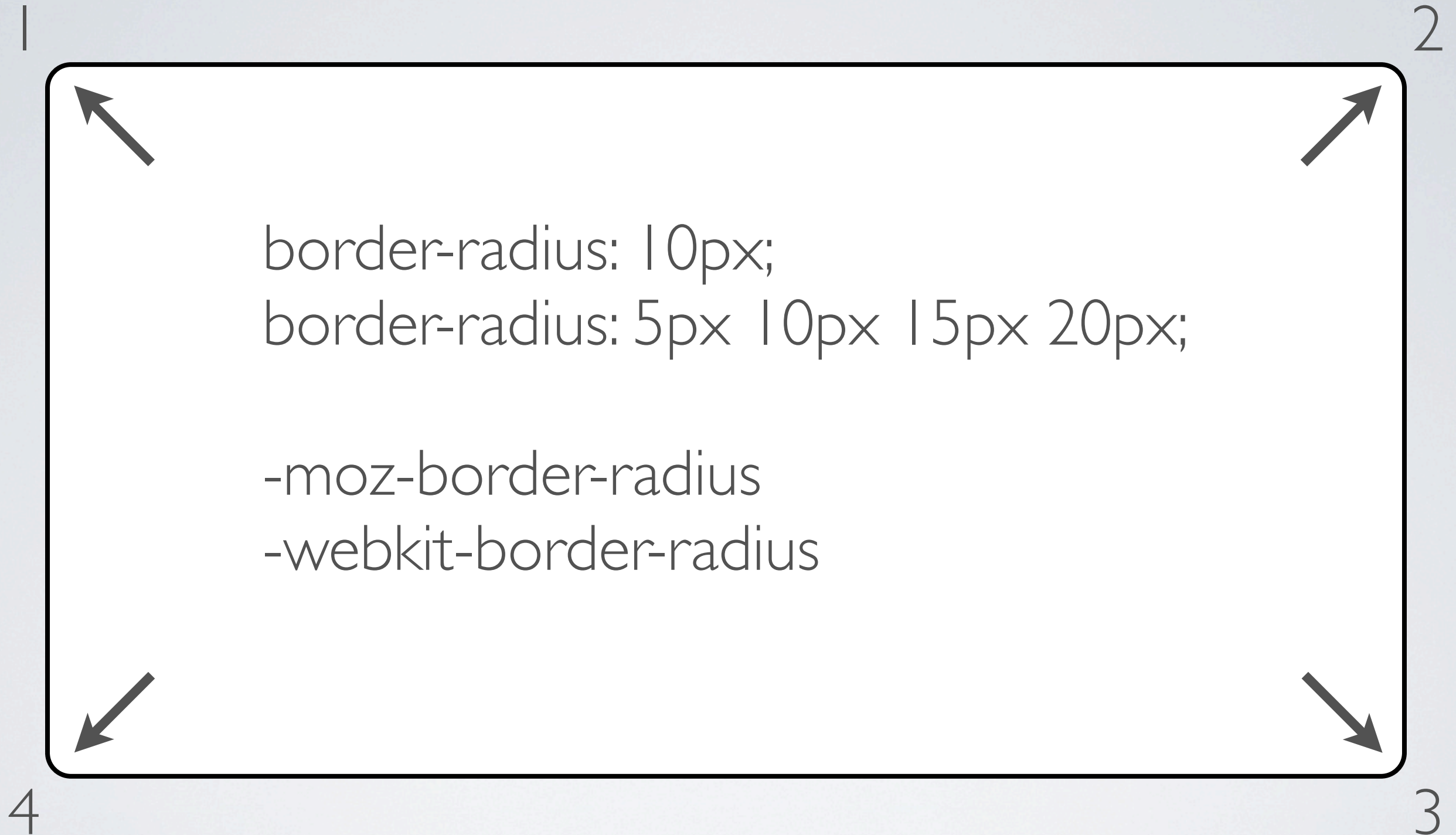
Utilize o bom senso



Temos vontade de matar o designer quando ele pede para...

Temos vontade de matar o designer quando ele pede para...

- ▶ Fazer caixinhas com bordas arredondadas
- ▶ Colocar uma sombrinha nos textos/caixas
- ▶ Utilizar um background gigante
- ▶ Utilizar um background composto
- ▶ Fazer uma página adaptável ao tamanho da tela
- ▶ Fazer os títulos das páginas ter uma fonte mais “style”





```
box-shadow: inset 5px 5px 5px #ff0000;
```

```
-moz-box-shadow  
-webkit-box-shadow
```

```
box-shadow: 10px 10px 10px #000000;
```



Qual o problema do background gigante?

Qual o problema do background gigante?



Qual o problema do background gigante?

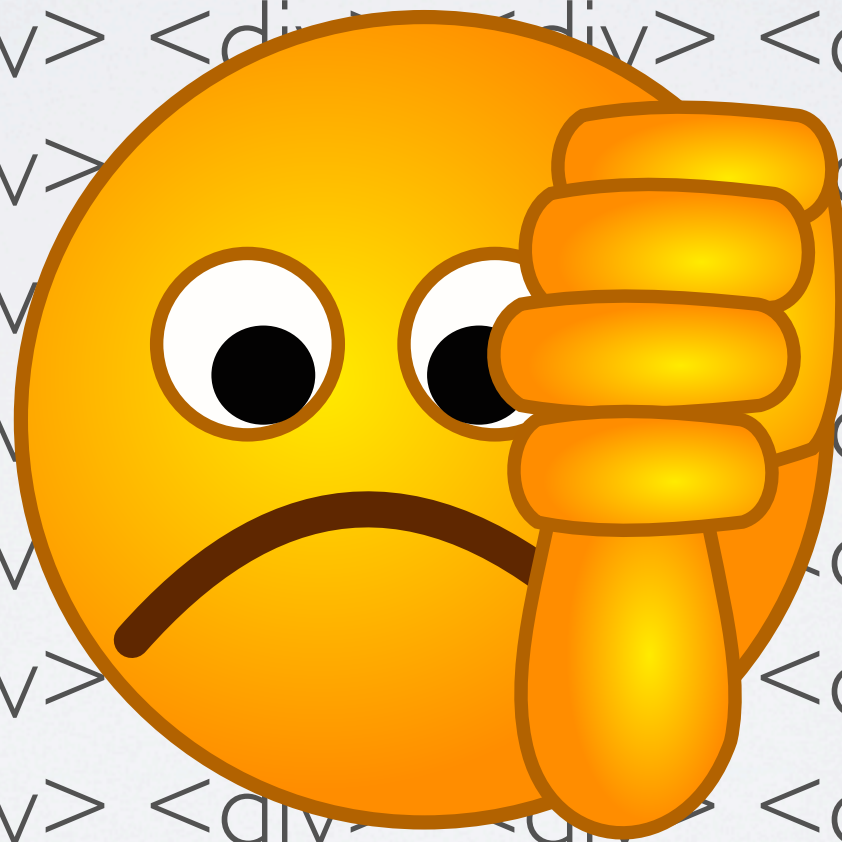
Solução:

SVG + background-size

Como resolvemos hoje o problema dos backgrounds compostos?

Como resolvemos hoje o problema dos backgrounds compostos?

<div> <div> <div> <div> <div> <div> <div> <div>
<div> <div> <div> <div> <div> <div> <div> <div>
<div> <div> <div> <div> <div> <div> <div> <div>
<div> <div> <div> <div> <div> <div> <div> <div>
<div> <div> <div> <div> <div> <div> <div> <div>
<div> <div> <div> <div> <div> <div> <div> <div>
<div> <div> <div> <div> <div> <div> <div> <div>
<div> <div> <div> <div> <div> <div> <div> <div>
<div> <div> <div> <div> <div> <div> <div> <div>



Como resolvemos hoje o problema dos backgrounds compostos?

Solução:

Múltiplas declarações de background.

Como resolvemos o problema das páginas com aparências diferentes para tamanhos diferentes de tela?

Como resolvemos o problema das páginas com aparências diferentes para tamanhos diferentes de tela?

Antes do CSS3:

```
@media screen { p { font-family: sans-serif } }
```

```
<link rel="stylesheet" media="print" href="..." />
```


Como resolvemos o problema das páginas com aparências diferentes para tamanhos diferentes de tela?

Antes do CSS3:

```
@media screen { p { font-family: sans-serif } }  
<link rel="stylesheet" media="print" href="..." />
```

CSS3:

```
@media all and (min-width:500px) { ... }  
@media print and (orientation: portrait) { ... }  
<link rel="stylesheet" media="print and (min-width:  
25cm)" href="..." />
```

É as fontes diferentes das que não são as do padrão web?

E as fontes diferentes das que não são as do padrão web?

Como era feito até hoje?

Imagem estática

SIFR (Scalable Inman Flash Replacement)

Cufón

Imagem dinâmica (ex: FLIR)

E as fontes diferentes das que não são as do padrão web?

Como era feito até hoje?

Imagem estática

SIFR (Scalable Inman... nt)

Cufón

Imagem dinâmica (ex



E as fontes diferentes das que não são as do padrão web?

Com CSS3 fica assim:

```
@font-face {  
font-family: nomeDaFonte;  
src: url('/local-da-fonte/arquivo-da-fonte') format("tipo-da-fonte");  
}
```

Na hora de usar:

```
ELEMENTO {  
font-family: nomeDaFonte, verdana, helvetica;  
}
```

Padrões de formatos de fonte:

WOFF (webfonts)

TTF (Truetype)

OTF (Opentype)

SVG (Scalable Vector Graphic) → iPhone/iPad/WebKit

Novos seletores de atributo:

`E[foo^="bar"]`

`E[foo$="bar"]`

`E[foo*="bar"]`

Seletor de “irmão” genérico:

$E \sim F$

Funciona exatamente como o $E + F$, porém F não precisa estar imediatamente precedido de E

Novas pseudo-classes:

E:root

E:nth-child(n)

E:nth-last-child(n)

E:nth-of-type(n)

E:nth-last-of-type(n)

E:last-child

E:first-of-type

E:last-of-type

E:only-child

E:only-of-type

E:empty

E:enabled

E:checked

E:not(s)



That's all Folks!

HTML5 & CSS3

Jonas Keizo Hirata



www.k19.com.br

twitter: [@k19treinamentos](https://twitter.com/k19treinamentos)

contato@k19.com.br