

Web Design

Responsivo e Adaptativo

Heraldo G. Lima Junior
Web designer - NTI | FACAPE
Estagiário



Construindo Layouts Tableless

- Layouts tableless são layouts ou páginas web construídas e estruturadas sem o uso das antigas tabelas (`<table></table>`), substituindo-as pela tag `<div>` para montar os “blocos” do layout.

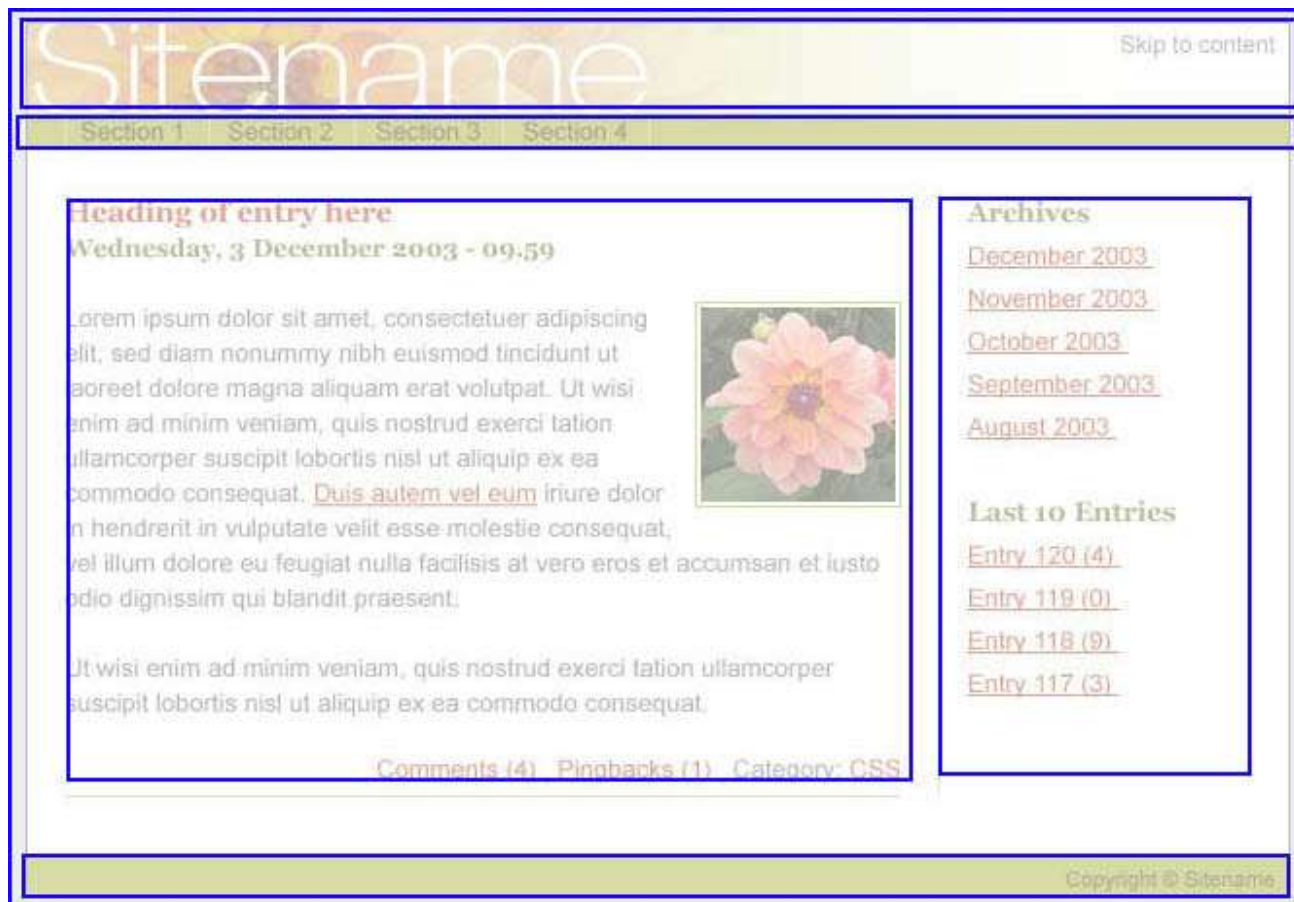


Obs1.: Desenvolva para os navegadores modernos e depois adapte para os antigos

Construindo Layouts Tableless

#1º Passo

> Escolha os “blocos” do layout.



Construindo Layouts Tableless

#2º Passo

> Nomear os blocos. Se o bloco for único, utilize **ID** no lugar de **CLASSE**, pois caso ocorra conflito de regras css, o seletor ID tem prioridade sobre o seletor CLASSE.



Construindo Layouts Tableless

#3º Passo

> Criar a marcação.

Comece a declaração de DOCTYPE:

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

Prossiga com a seção **head** do documento, incluindo a codificação de caracteres, link para folhas de estilo:

```
<head>  
<meta http-equiv="content-type" content="text/html; charset=utf-8">  
<title>Page title</title>  
<link rel="stylesheet" href="style.css" type="text/css" media="screen">  
</head>
```

Construindo Layouts Tableless

#4º Passo

> Crie as divs para estruturar o seu layout, logo após a tag <body>:

```
<body>
  <div id="container">
    <div id="header"></div>
    <div id="mainnav"></div>
    <div id="menu"></div>
    <div id="contents"></div>
    <div id="footer"></div>
  </div>
</body>
</html>
```

Construindo Layouts Tableless

#5º Passo

> Posicionar os blocos:

```
body {  
margin: 0; padding: 0; background: #ddd;  
}  
#container {  
margin: 1em auto; width: 650px; background: #fff;  
}  
#header {background: #CF3;}  
#mainnav {background: #9F3; }  
#menu {  
float: right; width: 165px; background: #6F9;  
}  
#contents {  
float: left; width: 440px; background: #9FC; margin: 0 0 0 20px;  
}  
#footer {  
clear: both; background: #FF9;  
}
```

Construindo Layouts Tableless

#6º Passo

> Adicione conteúdo aos blocos.



e...

Construindo Layouts Tableless

#Menu Dropdown

> Lógica básica:

Posiciona relativamente (position:relative) e flutua à esquerda (float:left).

** **

Esconda (display:none) e posicione absolutamente (position:absolute).
E quando o mouse passar sobre o

o submenu é mostrado (display:block).

Construindo Layouts Tableless

#Menu Dropdown

> HTML:

```
<ul>
  <li><a href='#>Sub-íten 1</a>
    <ul>
      <li> <a href='#>Íten 2</a> </li>
      <li> <a href='#>Íten 3</a> </li>
    </ul>
  </li>
  <li><a href='#>Sub-íten 2</a></li>
  <li><a href='#>Sub-íten 3</a></li>
</ul>
```

Construindo Layouts Tableless

#Menu Dropdown

> CSS:

```
.menu li { position: relative; float: left; list-style: none; height: 50px; width: 100px; border: 1px solid #b3b3b3; text-align: center; text-shadow: 1px 1px 2px #fff; }
```

```
.menu li a { display: block; background: -webkit-linear-gradient(#b3b3b3, #fff); color: #333; text-decoration: none; font-family: arial; height: 35px; padding-top: 15px; }
```

```
.menu li a:hover { background: -webkit-linear-gradient(#000, #333); color: #fff; text-shadow: 0px 0px 10px #fff; }
```

```
.menu li ul { position: absolute; left: -40px; display: none; }
```

```
.menu li ul li { list-style: none; height: 50px; width: 150px; border: 1px solid #b3b3b3; text-align: center; }
```

```
.menu li:hover ul { position: absolute; left: -40px; display: block; }
```

Construindo Layouts Tableless

#Menu Dropdown

> Resultado:



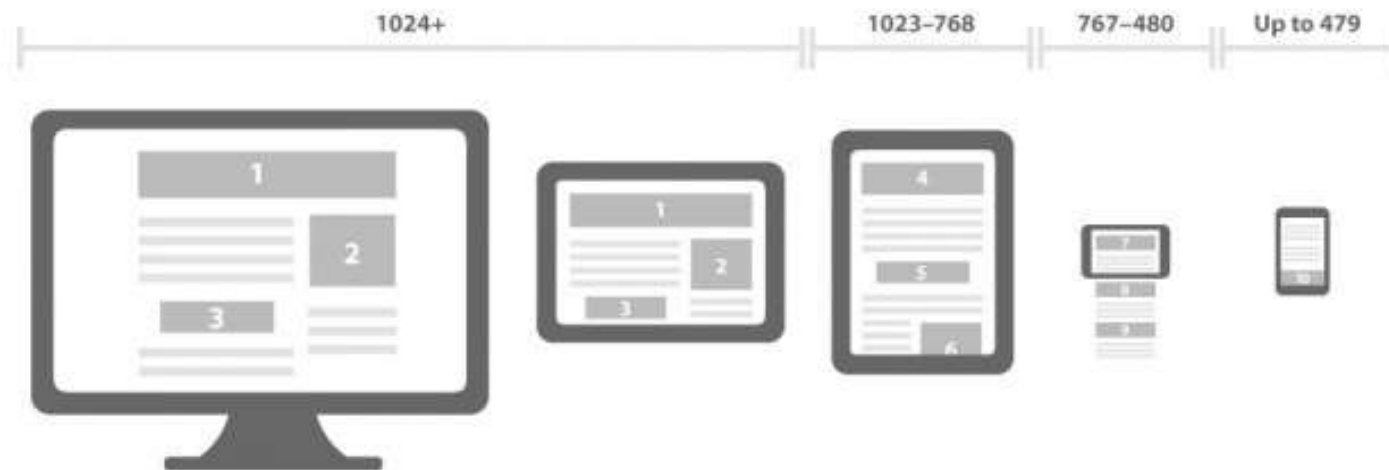


**# Seu layout
está pronto!**

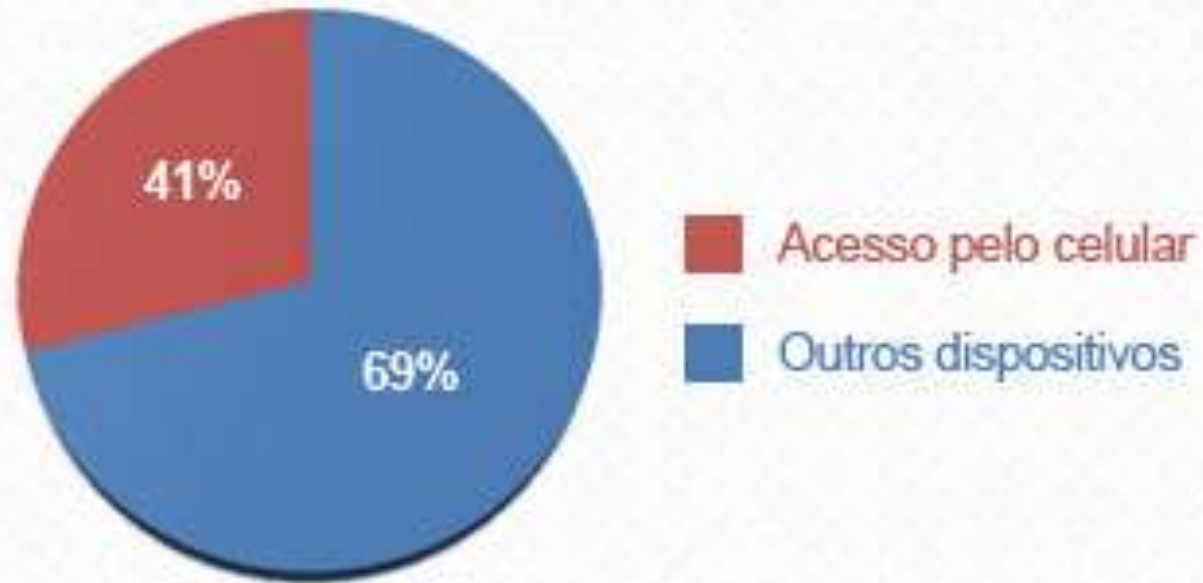
Schwaminoff

O que é Web Design Responsivo?

- É a técnica de criar páginas para a web que se adaptem a diferentes resoluções, telas, dispositivos, sem precisar criar uma página específica para cada situação, mantendo sempre a acessibilidade e a usabilidade da página.



Acesso à internet em 2012



Fonte: Webop - <http://webop.com.br/blog/mobile/o-crescimento-da-internet-movel-no-brasil>

Usuários satisfeitos!

Aplicações

Eficazes

Eficientes

Bonitas

Fáceis de usar

Em qualquer DISPOSITIVO!



Usabilidade / Acessibilidade

- > **Dica 1**

- ❖ **CSS alternativo para navegadores antigos:**

É impossível desenvolver um css que renderize em todos os navegadores. Para resolver este problema se utiliza de css's alternativos.

Ex.:

```
<!--[if IE]>
```

```
Aqui podemos informar o código que quisermos.
```

```
<![endif]-->
```

Usabilidade / Acessibilidade

* CSS alternativo para navegadores antigos

Ex.:

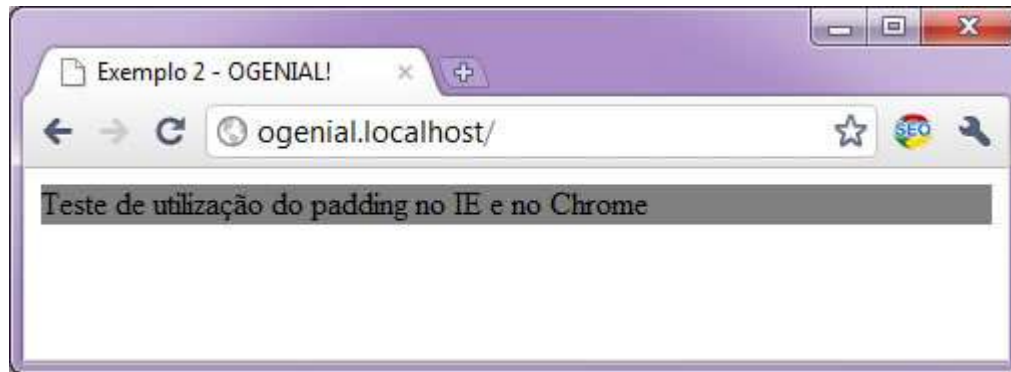
```
<html>
  <head>
    <style type="text/css">
      #main {
        background-color: gray;
      }
    </style>
    <!--[if IE]>
      <style type="text/css">
        #main {
          padding: 10px;
        }
      </style>
    <![endif]-->
  </head>
  <body>
    <div id="main">
      Teste de utiliza&ccedil;&atilde;o do padding no IE e no Chrome
    </div>
  </body>
</html>
```

O código ao lado aplica um css diferente para o IE. Ao abrir este arquivo no IE, a div de id="main" receberá um *padding* de 10px, enquanto nos demais navegadores ela ficará sem *padding*.

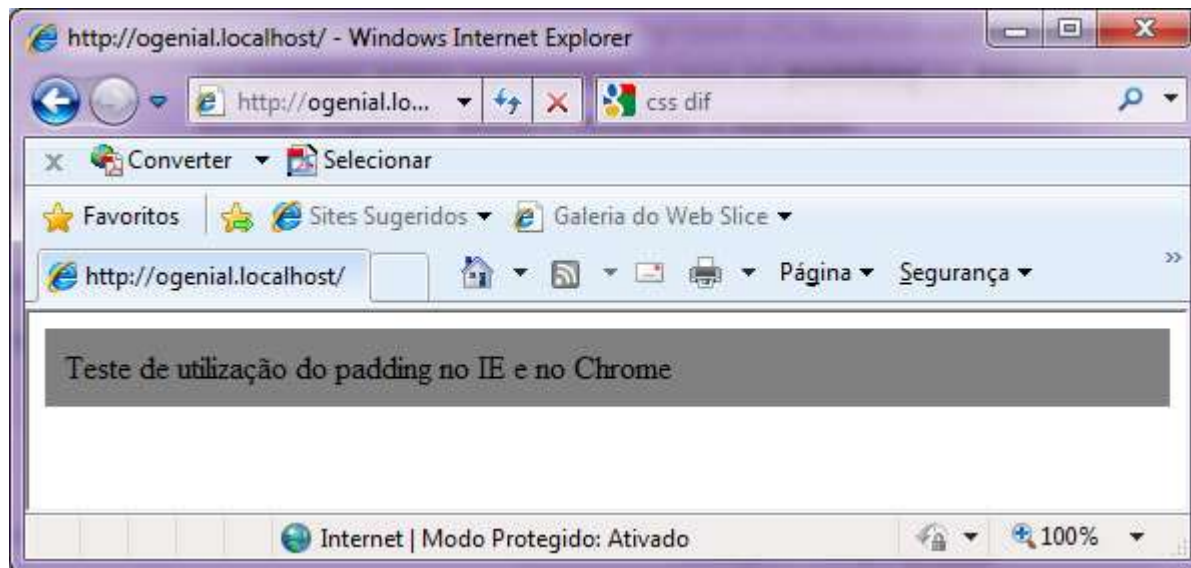
Usabilidade / Acessibilidade

* CSS alternativo para navegadores antigos

Resultado da aplicação do código anterior::



Google Chrome.



Internet Explorer.

Usabilidade / Acessibilidade

* CSS alternativo para navegadores antigos

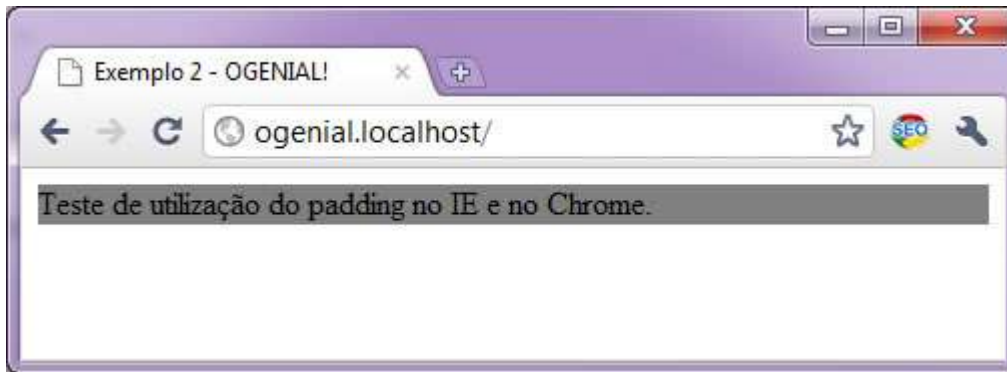
Dentro das tags <body></body>:

```
<body>
  <!--[if IE]>
    <div id="ie">
      Baixe um navegador mais moderno, utilize Google Chrom
e, ou Firefox.
    </div>
  <![endif]-->
  <div id="main">
    Teste de utiliza&ccedil;&atilde;o do padding no IE e no
Chrome.
  </div>
</body>
```

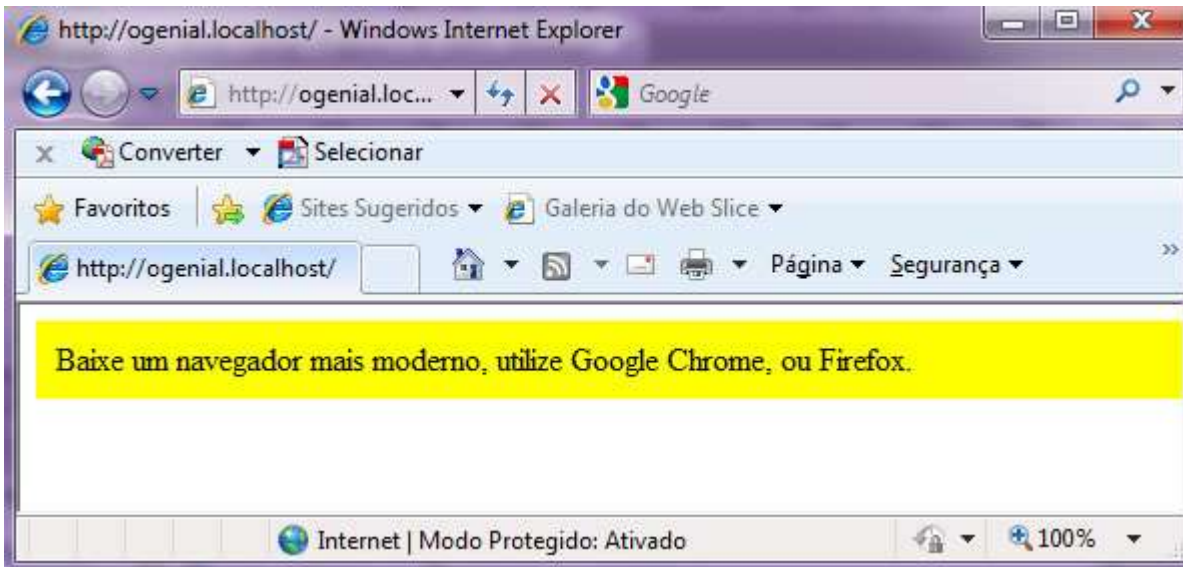
Usabilidade / Acessibilidade

* CSS alternativo para navegadores antigos

Resultado:



Google Chrome.



Internet Explorer.

Usabilidade / Acessibilidade

*Media Queries

Imagens Fluídas:

As imagens passam a se adaptar de acordo com o tamanho do layout, basta apenas inserir a linha de código a seguir:

```
img {  
    max-width: 100%;  
}
```

Usabilidade / Acessibilidade

*Media Queries

Imagens Fluídas:

Outra técnica bastante útil é cortar as laterais da imagem, fazendo com que ela fique contida na div a qual pertence, fazendo com que partes da imagem escondam-se ou apareçam de acordo com a resolução da tela.

```
.div_contetora_da_img {  
    overflow: hidden;  
}
```

Media Queries



Usabilidade / Acessibilidade

*Media Queries

Especificam um estilo específico de acordo com a media, resolução, largura, etc. Os mais utilizados são:

❖all

Para todos os dispositivos.

❖handheld

Para dispositivos de mão. Normalmente com telas pequenas e banda limitada.

❖print

Para impressão em papel.

❖projection

Para apresentações, como PowerPoint.

❖screen

Para monitores ou outros dispositivos com telas coloridas e com resolução adequada.

❖tv

Para dispositivos como televisores, ou seja, com baixa resolução, quantidade de cores e scroll limitado.

Usabilidade / Acessibilidade

*Media Queries

- Seu uso é feito dentro das tags `<head>` e `</head>`.

```
<link rel="stylesheet" href="estilo.css" media="screen and (color)" />
```

- Ou in-line:

```
@media print {  
    /* estilos */  
}
```

Usabilidade / Acessibilidade

*Media Queries

- **Parametros do Media Queries:**

- **Width** - Largura do viewport (janela do browser).
- **Height** - Altura do viewport (janela do browser).
- **Device** - width - Largura da mídia.
- **Device** - height - Altura da mídia.
- **Orientation** – Orientação da Mídia.
- **aspect-ratio** – Proporção.
- **device-aspect-ratio** - Proporção da tela do dispositivo.
- **Color** - Número de bits por cor.
- **color-index** - Número de entradas na tabela de pesquisa de cores.
- **Monochrome** - Número de bits por pixel em um buffer de quadros monocromáticos.
- **Resolution** - Resolução do dispositivo.
- **Scan** - Tipo de formação de imagens específico para televisores.
- **Grid** - Determina se o dispositivo é baseado em bitmap ou em um grid.

Usabilidade / Acessibilidade

*Media Queries

- Operadores Lógicos:

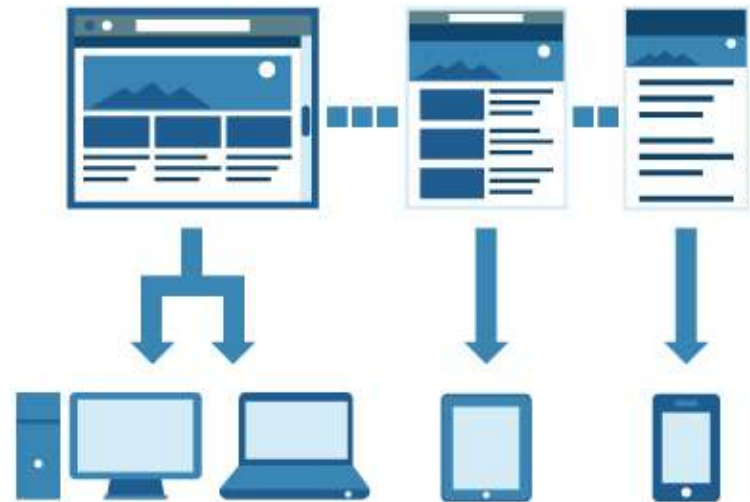
- **Ou, And e Only.**

- ✓ Se utiliza o only para dizer ao navegador que o arquivo css informado só será carregado em navegadores que suportam media queries.

Usabilidade / Acessibilidade

*Media Queries

- Principais resoluções utilizadas:
- **320 pixels** - Smartphones no modo retrato.
- **480 pixels** - Smartphones no modo paisagem.
- **600 pixels** - Tablets pequenos. Ex: Amazon Kindle (600×800)
- **768 pixels** - Tablets maiores em modo retrato. Ex: iPad (768×1024)
- **1024 pixels** - Tablets maiores em modo paisagem, monitores antigos.
- **1200 pixels** - Monitores wide.



Usabilidade / Acessibilidade

*Media Queries

Exemplos:

Um iPhone em modo retrato, por exemplo, possui 320px de width. Se você desenvolver um CSS para o smartphone da Apple basta utilizar o seguinte Media Querie:

```
/* Smartphone em modo retrato */  
@media only screen and (max-width : 320px) {  
    /* estilos */  
}
```

Usabilidade / Acessibilidade

*Media Queries

Exemplos:

Ipad, independente da orientação:

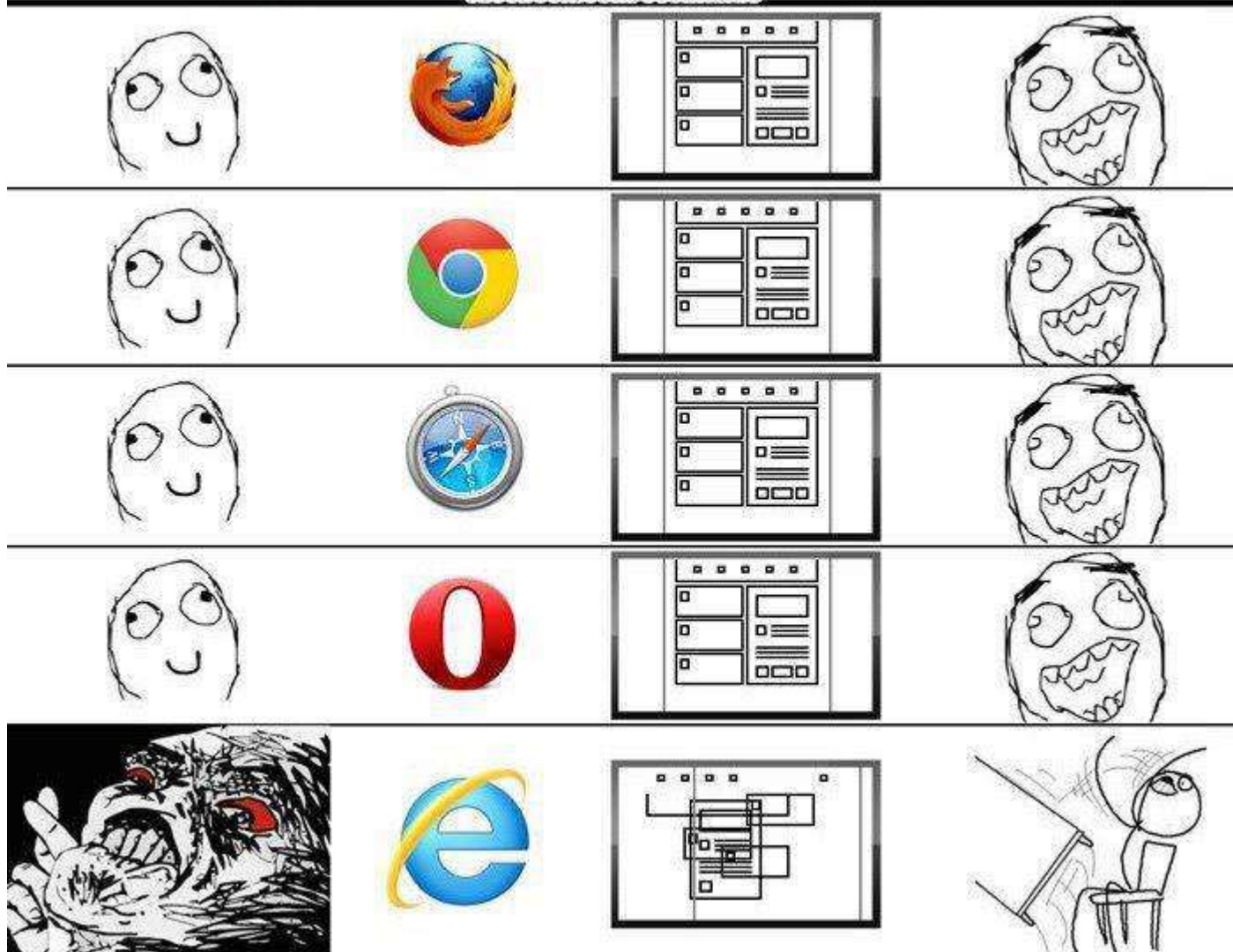
```
/* iPads (retrato e paisagem) */  
@media only screen  
and (min-device-width : 768px)  
and (max-device-width : 1024px) {  
    /* estilos */  
}
```

Mas afinal,
funciona em
todos os
browsers?



TESTANDO DIFERENTES NAVEGADORES

facebook.com/ovelhando





Calma, para tudo tem-se um jeito...

Usabilidade / Acessibilidade

*Media Queries

➤css3-mediaqueries-js ()

Download em:

<http://code.google.com/p/css3-mediaqueries-js/>

Usabilidade / Acessibilidade

*Media Queries

Testando seu site em dispositivos mobile:



WHY GO MO? **TEST YOUR SITE** BUILD YOUR SITE SPREAD THE WORD GO MO GUIDES

SEE WHAT YOUR CUSTOMERS SEE

This tool shows you how your current site looks on a smartphone, and provides a report on what's working and what you can do better. To get started, fill in your info below.

TEST YOUR SITE



Disponível em:

<http://www.howtogomo.com>

HTML

CSS



CSS



Principais novidades

- Gradiente em textos e elementos;
- Bordas arredondadas;
- Sombras em texto e elementos;
- Manipulação de opacidade;
- Controle de rotação;
- Controle de perspectiva;
- Animação;
- Estruturação independente da posição no código HTML;



Adeus
Photoshop!



Gradientes

```
div {  
  width:200px;  
  height:200px;  
  background-color: #FFF;
```



```
/* imagem caso o browser não aceite a feature */  
background-image: url(images/gradiente.png);  
/* Firefox 3.6+ */  
background-image: -moz-linear-gradient(green, red);  
/* Safari 5.1+, Chrome 10+ */  
background-image: -webkit-linear-gradient(green, red);  
/* Opera 11.10+ */  
background-image: -o-linear-gradient(green, red);  
}
```

Menos Java Script...



Transições

Transition:

```
a {  
  color: white;  
  background: gray;  
  -webkit-transition: 0.5s;  
}  
a:hover {  
  color: black;  
  background: red;  
  -webkit-transition: 0.5s;  
}
```

Proporciona uma transição mais suave durante a mudança de um estado para outro.

Bordas

Bordas Arredondadas:

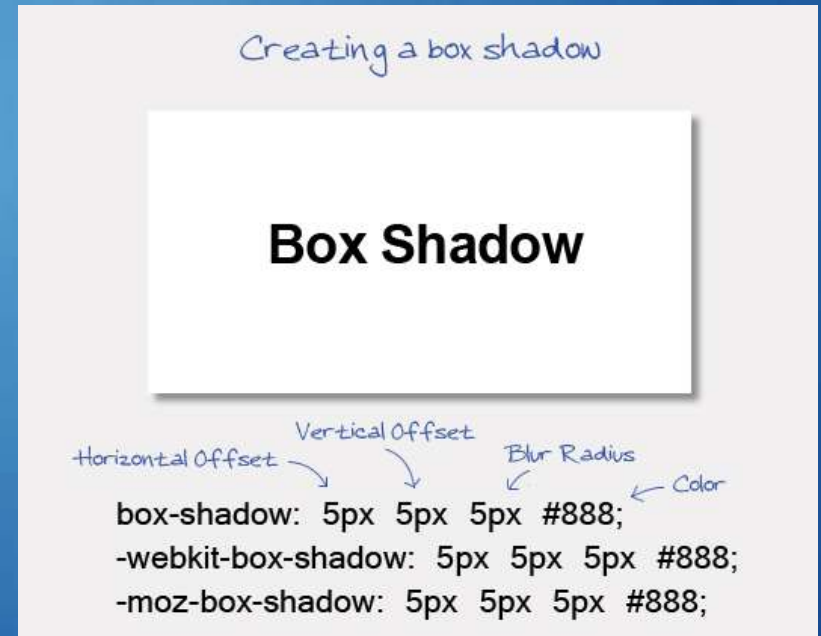
```
div {  
    height: 50px;  
    width: 100px;  
    background: black;  
    border: 5px solid yellow;  
    border-radius: 20px;  
}
```



Sombras

Caixas:

```
div {  
  height: 50px;  
  width: 100px;  
  background: black;  
  border: 5px solid yellow;  
  border-radius: 20px;  
  box-shadow: 0px 0px 10px #000;  
}
```



Sombras

Texto:

```
p {  
  font-size: 50px;  
  color: #fff;  
  font-family: arial, sans-serif;  
  Text-shadow: 0px 0px 10px #000;  
}
```

3D CSS TEXT
WITH 'ZOOM'

HTML



Validando Formulários

- Em HTML5 existem atributos que validam o formulário no momento em que o usuário submeter os dados, como o **min, max, pattern, step, required, type e maxlength**.
- **Compatível apenas com navegadores novos.**

Required

- Mostra que o campo é de preenchimento obrigatório.
- Ex.:

```
<form>  
  <input type="text" required value="" />  
  <input type="submit" value="Submit" />  
</form>
```

Required

Chrome



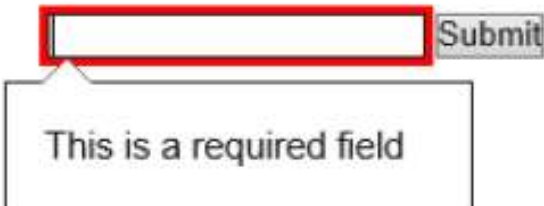
A text input field with a blue border and a "Submit" button. Below the field, a white error message box with a yellow warning icon and the text "Please fill out this field." is displayed.

Firefox



A text input field with a blue border and a "Submit" button. Below the field, a dark grey error message box with the text "Please fill out this field." is displayed.

IE10



A text input field with a red border and a "Submit" button. Below the field, a white error message box with the text "This is a required field" is displayed.

Opera



A text input field with a blue border and a "Submit" button. Below the field, a pink error message box with the text "This is a required field" is displayed.

Pattern

- Determinamos o padrão de preenchimento do campo por meio de uma expressão regular.
- Ex.:

```
<input pattern="[0-9]{3}[A-Z]{3}" />
```

Min e Max

- Determinamos os valores mínimos e máximos para os campos.
- Ex.:

```
<input type="number" max="10" />
```

Min e Max

- Determinamos os valores mínimos e máximos para os campos.
- Ex.:

```
<input type="number" max="10" />
```

```
<input type="number" max="9" min="1" />
```

```
<input type="date" min="1979-12-31" />
```

Step

- Determina um intervalo de números e uma determinada sequência.
- Ex.:

```
<input type=number min="0" max="10" step="0.5">
```

Personalizando a mensagem de erro.

- Para isso se utiliza o atributo **title**.
- Ex.:

```
<input type=number min="1" max="10"  
  title="Preencha o campo corretamente!" />
```

*Não substitui a mensagem padrão, aparece junto com ela.

As pseudo-classes :valid e :invalid

- Estilizam o formulário que foi preenchido de forma válida (:valid) e os que estão preenchidos de forma errada (:invalid).
- Ex.:

```
input:invalid { border: 1px solid red; }
```


Referências

- ❑ MAUJOR. **Layout CSS passo a passo**. Disponível em <<http://www.maujor.com/tutorial/layout-css-passo-a-passo.php>> Acesso em março de 2013.
- ❑ SCHIRM, Lucas. **Como utilizar um CSS diferente para cada navegador**. Disponível em <<http://www.ogential.com.br/blog/css-diferentes-navegador/>> Acesso em março de 2013.
- ❑ GUERRATO, Dani . **Design Responsivo III – Media Queries e Compatibilidade**. Disponível em <<http://blog.popupdesign.com.br/design-responsivo-iii-media-queries-e-compatibilidade/>> Acesso em março de 2013.
- ❑ W3C. **CSS. Curso W3C Escritório Brasil**.
- ❑ ALTERMANN, Dennis. **Design Responsivo: Entenda o que é a técnica e como ela funciona**. Disponível em <<http://midiatismo.com.br/o-mobile/design-responsivo-entenda-o-que-e-a-tecnica-e-como-ela-funciona>> Acesso em março de 2013.
- ❑ ALBAN, Afonso. **Ampliando a usabilidade de interfaces web para idosos em dispositivos móveis: uma proposta utilizando design responsivo**. Passo Fundo: *CINTED-UFRGS*, 2012.
- ❑ EIS, Diego. **Introdução sobre Media Queries**. Disponível em < tableless.com.br/introducao-sobre-media-queries/#.UUyTgxykqWA > Acesso em março de 2013.
- ❑ W3C. **Técnicas CSS para acessibilidade a conteúdo web - Diretrizes 1.0**. Disponível em < <http://www.w3.org/TR/2000/NOTE-WCAG10-CSS-TECHS-20001106/> > Acesso em março de 2013.

Obrigado pela Atenção.