

HTML5 & CSS3

Por Marta Preuss baseado no Codeshow da Visie

versão 1.1

Colaboradores com correções e adendos: @rafasoaes, @lentu e @garotageek

2010

Índice

HTML	3
<i>Por que HTML5?</i>	3
<i>Já dá pra usar HTML5 hoje em dia?</i>	3
<i>Quais são as diferenças?</i>	4
Doctype	4
HTML	4
Charset	4
Chamando CSS e JS	5
Imagem	5
Video	6
<i>Tags novas do HTML5 para montagem da página</i>	7
<i>Estrutura semântica da tabela</i>	8
<i>Estrutura semântica de formulários</i>	9
<i>Adaptando para celulares</i>	10
<i>Dicas de desenvolvimento</i>	11
CSS3	12
Referência, dicas, links úteis	13

HTML

Por que HTML5?

Na opinião de Diego Eis, da Visie, o HTML5 vem para substituir o HTML4, depois que o xHTML teve seu desenvolvimento suspenso. O desenvolvimento do xHTML no W3C estava muito devagar, então um grupo de desenvolvedores começou o HTML5 e fez tanto barulho que o W3C pegou para si. Tem sido lançado devagar e liberado aos poucos, e assim sendo implantado pelos browsers.

Com o HTML5 a semântica é quase obrigatória. Com isso, passar o site para mobile é extremamente simples e ganhamos muito tempo na manutenção.

Já dá pra usar HTML5 hoje em dia?

SIM. A única coisa que o browser faz é interpretar seja lá o que for que você escrever e formatar com CSS. Como o IE não é browser, ele precisa de um javascript que transforma as caixas em divs.

É basicamente um

```
<script>document.createElement('time');</script>
```

Onde 'time' é a tag do HTML5 que você quer transformar em caixa.

Existem scripts que fazem isso automaticamente, é só adicionar no <header>. Esse está hospedado no Google, é bom que não consome banda.

```
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
```

O que não dá pra usar ainda são muitas das frescuras do CSS3. Bordas arredondadas, colunas, sombras são elementos que já funcionam no webkit e no firefox mas ainda não existem para IE e algumas coisas nem pro Opera. Isso acontece porque, como está se implementando aos poucos, os navegadores fazem algumas coisas e outros não. Existem tabelas que mostram o que funciona aonde. As opções são ou contornar o problema com jQuery ou conversar com o cliente, oferecendo uma versão com features de layout para quem usa browsers modernos.

Quais são as diferenças?

Não é mais necessário fechar todas as tags, uma vez que não estamos mais no xHTML. De `
` fica `
` mesmo.

Doctype

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

Para
`<!DOCTYPE html>`

HTML

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

Para
`<html lang="pt-BR">`

É importante colocar o atributo lang porque isso ajuda o google a indexar, configura o leitor de tela e ainda ajuda o charset.

Charset

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

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

Uma vez colocado, não é preciso se preocupar em colocar caracteres com ASCII, tipo ` `; . Se você fixa o caracter com o código ASCII, numa tradução para caracteres diferentes esse código permanece igual. Isso prejudica a acessibilidade.

Chamando CSS e JS

De

```
<style type="text/css">
```

Para

```
<style>
```

De

```
<script type="text/javascript">
```

Para

```
<javascript>
```

Imagem

De

```
<img src="" alt=""/>
```

```
<span> Legenda </span>
```

Para

```
<figure>
```

```
  
```

```
  <figcaption>A cheeky macaque, Lower Kintaganban River, Borneo. Original by <a href="http://  
www.flickr.com/photos/rclark/">Richard Clark</a></figcaption>
```

```
</figure>
```

No W3Schools diz ainda que não existem mais os atributos align, border, hspace, e vspace.

Video

Antes

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" width="425" height="344"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,40,0">
<param name="allowFullScreen" value="true" />
<param name="allowscriptaccess" value="always" />
<param name="src" value="http://www.youtube.com/v/oHg5SJYRHA0&hl=en&fs=1&" />
<param name="allowfullscreen" value="true" />
<embed type="application/x-shockwave-flash" width="425" height="344" src="http://www.youtube.com/v/
oHg5SJYRHA0&hl=en&fs=1&" allowscriptaccess="always" allowfullscreen="true">
</embed>
</object>
```

Depois

```
<video width="320" height="240" controls="controls">
  <source src="movie.ogg" type="video/ogg">
  <source src="movie.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
```

Existem dois tipos de codificação de vídeo: MP4 que funciona para todos os browsers menos o Firefox e o OGG. O MP4 roda inclusive no IE (embedando no flash), então a única preocupação é fazer as duas versões.

Vale lembrar que também existe o codec VP8 (da On2, que foi comprada pelo Google), que será usado no IE9, Firefox, Safari, Chrome e Opera.

Isso é uma coisa que precisamos estudar melhor. Ficou na sala de programação do Codeshow e eu não vi.

Tags novas do HTML5 para montagem da página

No lugar de usar `<div id="">` para tudo, vamos usar `div` quando for realmente um bloco de alguma coisa.

Atenção: as tags listadas abaixo são inline por default. Lembre-se de trocar o `display` para `block` no CSS.

`<header>` Coisas do cabeçalho, como título, busca, etc.

`<nav>` Para navegação interna do site.

`<aside>` Barra lateral.

`<section>` Um bloco, um pedaço.

`<article>` Um artigo. O conteúdo em si.

`<time>` Para data e hora.

`<footer>` Rodapé da página. Eu também vi usando como rodapé do artigo. Acho interessante, podemos discutir e usar a `header` para isso também.

Existem outras tags mas essas são as principais.

Estrutura semântica da tabela

Válido também para xHTML.

```
<table>
  <thead>
    <tr>
      <th>titulo</th>
      <th>Views</th>
      <th>x</th>
    </tr>
  </thead>
  <tfoot>
    <tr><td></td></tr>
  </tfoot>
  <tbody>
    <tr><td></td></tr>
  </tbody>
</table>
```

O `<thead>` é o cabeçalho, onde vai o que é cada coluna. Importante para acessibilidade e leitores de tela.

O `<tfooter>` vem logo abaixo por causa dos tempos negros de ADSL, onde o browser carregava o cabeçalho, o rodapé e só depois o meio da tabela.

`<tbody>` é onde vai o conteúdo.

Estrutura semântica de formulários

Válido também para xHTML

```
<form action="" method="POST" id="sampleForm">
  <fieldset><legend>Formulário</legend>
    <label for="userName">
      Nome de Usuário:
      <input type="text" id="userName" name="userName">
    </label>
    <label for="mail">
      Email:
      <input type="text" id="mail" name="mail">
    </label>
    <label for="fullName">
      Nome Completo:
      <input type="text" id="fullName" name="fullName">
    </label>
    <label for="choose">
      Escolha:
      <select name="choose" id="choose" >
        <option>asdaasdasdaafsd fsdfsds</option>
        <option>123123123123</option>
      </select>
    </label>
    <label for="address">
      Endereço:
      <textarea name="address" rows="3" cols="50" id="address" ></textarea>
    </label>
    <input type="submit" class="submit">
  </fieldset>
</form>
```

O `<fieldset>` é importante para acessibilidade e você pode tirar a borda, formatando o `<legend>` para ser o título do formulário, sem necessidade de `<hx>`.

Colocar o nome do campo e o campo dentro do `<label>` ajuda a clicar, pois clicando no título do campo já foca o mesmo.

Nota: No ASP.NET MVC, a função `Html.TextBox("campo")` - que substitui o `<asp:Textbox />` - renderiza um `<input type="text" name="campo" id="campo">` (e o mesmo vale para todas as funções para campos de formulários)

Adaptando para celulares

A tag mais coisa-linda-de-Deus (Diego Eis que descreveu assim)

```
<meta name="viewport" content="width=device-width" />
```

Que define a largura do dispositivo.

Com isso é possível fazer CSSs para cada largura:

```
<link rel="stylesheet" href="cssNormal.css" type="text/css" media="screen and (min-width:481px)">
```

```
<link rel="stylesheet" href="cssParaCelular.css" type="text/css" media="screen and (max-width:480px)">
```

Teste no Tableless.com.br, diminuindo a largura da tela. O CSS para mobile só tira os `float:left`.

O indicado é oferecer esse serviço para celulares mais novos, como o Android, os Nokias mais recentes e claro, iPhone.

Dica: esse `media="screen"` tem de ser em letra maiúscula para funcionar no windows mobile. Fica `media="Screen"`.

Dicas de desenvolvimento

Semântica não é algo mecânico. É algo a ser discutido e pensado - e, pessoalmente, é a parte mais divertida do HTML. No Codeshow mudávamos de ideia a cada sugestão, pensando para que servia cada coisa.

O melhor seria que planejássemos antes de montar, conversando com o gerente de projetos e o programador para saber o que é e para que serve cada elemento e, assim, escolher a tag mais adequada.

Outra coisa que é bacana é fazer todo o HTML primeiro, só a marcação, validar no W3C e só então passar a fazer o CSS. Enquanto isso, o programador pode já utilizar o HTML e assim toda a equipe ganha tempo.

Enfim, se fazemos todos os dias coisas muito parecidas, nada como um framework pra dar aquela ajuda. Já existem na internet frameworks prontos utilizando HTML5 e CSS3, mas o ideal é montarmos um de acordo com a nossa necessidade.

CSS3

Muitas das coisas do CSS3 ainda não vamos conseguir usar, mas seguem links de algumas dicas muito úteis que funcionam apenas em browsers com suporte:

Texto em colunas: <http://net.tutsplus.com/tutorials/html-css-techniques/quick-tip-the-multi-column-css3-module/>

Transitions, background na letra, rotate, box shadow, animation <http://www.smashingmagazine.com/2010/05/27/css-three-connecting-the-dots/>

Borda arredondada (inclusive na imagem), dois backgrounds na mesma div, sombra no texto <http://forum.imasters.uol.com.br/index.php?/topic/352661-o-css3-esta-chegando-confira-alguns-exemplos/>

Text gradient <http://net.tutsplus.com/tutorials/html-css-techniques/quick-tip-pure-css-text-gradients/>

A gente podia fazer um repositório com dicas sobre isso :)

Referência, dicas, links úteis

HTML5 em todos os browsers <http://html5doctor.com/how-to-get-html5-working-in-ie-and-firefox-2/>

Features you can use right now: [http://net.tutsplus.com/tutorials/html-css-techniques/quick-tip-html5-features-you-should-be-using-right-now/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed:+nettuts+\(NETTUTS\)](http://net.tutsplus.com/tutorials/html-css-techniques/quick-tip-html5-features-you-should-be-using-right-now/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed:+nettuts+(NETTUTS))

Formulários semânticos: <http://gaigalas.net/lab/semanticforms.html>

Tags HTML5 <http://www.w3schools.com/html5/>

Tag figure <http://html5doctor.com/the-figure-figcaption-elements/>

Lista de tags HTML <http://www.pinceladasdawe.com.br/html5/>

Tabela de compatibilidade HTML5 - browsers <http://html5demos.com/>

Exemplo da página em HTML5 - <http://www.pinceladasdawe.com.br/blog/2009/06/22/semantica-com-o-html5/>

Tabelas de HTML5 e CSS3 - <http://www.webresourcesdepot.com/html-5-and-css3-cheat-sheets-collection/>

Framework HTML5 e CSS3 - <http://www.52framework.com/>