



Programação em Ambiente Web I

Aula 6 – HTML5

Prof. Msc Cleyton Slaviero

`cslaviero@gmail.com`

Agradecimentos ao Prof. Jivago Medeiros (UFMT-Cba)
Fabiano Taguchi (UFMT-Roo)

HTML5

- O que conhecemos por HTML5 é a quinta, e atualmente última especificação da linguagem de marcação HTML.
 - É possível encontrar textos/documentos/livros que utilizam o termo HTML5 para se referirem ao conjunto HTML5+CSS3+JavaScript+Bibliotecas JS.

HTML5

- Especificação HTML é um esforço iniciado em 2004 e:
 - Define uma linguagem (HTML) que pode ser escrita em sua própria sintaxe e XML
 - Define modelos de processamento para promover implementações interoperáveis
 - Melhora as marcações/tags/elementos para documentos
 - Introduz marcações/tags/elementos e API's para "idiomas" emergentes, como em aplicações Web

HTML5

- HTML5 é compatível com HTML4 (não o contrário!)
 - Logo, poderemos ver por algum tempo páginas com HTML4 e 5 (até mesmo um site com HTML4 e 5)
 - Mas o contrário não é verdade!

HTML4 → HTML5

- Definição de codificação
 - HTML4
 - `<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">`
 - HTML5
 - `<meta charset="UTF-8">`
- Declaração do doctype
 - HTML4 – 3 tipos
 - `<!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">`
 - HTML5 - `<!DOCTYPE html>`

HTML5

- O HTML5 introduziu um maior número de “elementos semânticos” a linguagem HTML, como por exemplo:
 - O que seria um “elemento semântico” ?
 - É aquele elemento que possui um significado e conseqüentemente uma utilização explícita, que devem ser obedecidos, como por exemplo: ***section, header, article***.
 - Por que “um maior número” ?
 - Porque já haviam elementos semânticos no HTML4, por exemplo: ***form, img, table***, entre outros.
 - Então quais elementos não são semânticos?
 - Elementos como ***div*** e ***span*** não possuem nenhuma semântica enquanto elementos como ***p, b, i, u*** possuem pouca semântica.



HTML5

- Por que deve-se dar preferência a utilização de tag semânticas?
 - Auxiliam na organização e entendimento do documento
 - Permite que diferentes dispositivos e navegadores renderizem / manipulem os elementos de acordo com diferentes necessidades.

HTML5

- **<article>** : Define um conteúdo que geralmente contem textos e imagens, como por exemplo, uma postagem, uma notícia, um comentário, etc.
- **<aside>** : Define um conteúdo adjacente / ao lado de outro conteúdo.
- **<details>** / **<summary>** : Esse conjunto descreve um conteúdo com comportamento ocultar / mostrar. O conteúdo da tag *summary* será o conteúdo sempre visível.
- **<figure>** / **<figcaption>** : Encapsula imagem e adiciona uma legenda com **figcaption**
- **<footer>** : Conteúdo referente ao rodapé de um documento ou seção.
- **<header>** : Conteúdo referente ao cabeçalho de um documento ou seção.
- **<main>** : Tag que “encapsula” o conteúdo principal de um documento HTML.
- **<mark>** : Tag para destacar (highlight) um trecho de texto.
- **<nav>** : Define um conjunto de links de navegação (ou menus).
- **<section>** : Define uma seção do documento. Geralmente uma seção encapsula cabeçalhos, menus, artigos, etc
- **<time>** : Informações referentes a data e hora, como por exemplo a data de criação / publicação de um artigo



HTML4 → HTML5

HTML4

`<div id="header">`

`<div id="menu">`

`<div id="content">`

`<div id="post">`

`<div id="footer">`

HTML5

`<header>`

`<nav>`

`<section>`

`<article>`

`<footer>`

HTML5

- A utilização das tags introduzidas pelo HTML5 devem ser capaz de diminuir a “***div-dependência***” que “contaminou” muitos desenvolvedores web.
- Mas isso não significa que você “**nunca mais**” irá utilizar *tags* não semânticas, exmplo ***div*** e ***span***.
- Muitas vezes podem ser úteis para conseguirmos o posicionamento que queremos em um documento HTML.

HTML5

- MathML (<https://www.w3.org/Math/>)
 - Linguagem de marcação matemática
 - Objetivo é facilitar o uso e reuso de conteúdo matemático e científico
 - Tags XML para marcar uma equação quanto a semântica e apresentação

MathML: $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} e \\ f \end{pmatrix}$ and

another: $\begin{cases} ax + by = e \\ cx + dy = f \end{cases}$

HTML5

MathML

• $x^2 + 4x + 4 = 0$

```
<apply>
  <eq/>
  <apply>
    <plus/>
    <apply>
      <power/>
      <ci>x</ci>
      <cn>2</cn>
    </apply>
    <apply>
      <times/>
      <cn>4</cn>
      <ci>x</ci>
    </apply>
    <cn>4</cn>
  </apply>
  <cn>0</cn>
</apply>
```

```
<mrow>
  <mrow>
    <msup> <mi>x</mi> <mn>2</mn> </msup> <mo>+</mo>
    <mrow>
      <mn>4</mn>
      <mo>&InvisibleTimes;</mo>
      <mi>x</mi>
    </mrow>
    <mo>+</mo>
    <mn>4</mn>
  </mrow>
  <mo>=</mo>
  <mn>0</mn>
</mrow>
```

HTML5



- SVG (Scalable Vector Graphics)
 - Gráficos vetorizados
 - SVG é escrito em XML, logo tags devem ser fechadas
 - Imagens vetorizadas não perdem qualidade ao redimensionar a página

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

HTML5

- Elementos e atributos relacionados a formulário:
- ***input type:***
 - **color, date, datetime, number, range,** tel, search, url, email
- **Atributos:**
 - *autocomplete, autofocus, formaction, list, multiple, placeholder, required, min, max, step, required...*
- **Elementos:**
 - *datalist, keygen, output*
- **Atributo (e não tag) form**
 - Possibilita que um input possa ser colocado em qualquer lugar da página, e ainda fazer parte do form
 - `<form id=form_principal>...</form>`
 - `<input form=form_principal ... />`

Outros elementos novos

- Template
- Video
- Audio
- Track
- Embed
- Mark
- Progress
- Meter
- Time
- Ruby,rt,rp
- Bdi
- Wbr
- Canvas
- Datalist
- Keygen
- Output
- ...

<https://www.w3.org/TR/html5-diff/#backward-compatibility>

Elementos e atributos modificados

- Mudanças na semântica (significado) dos elementos e atributos
- Elementos
 - - não dá mais importância (strong sim)
 - <address> - é delimitado pelo body ou article
 - <i> - representa um pedaço de texto em uma voz alternada ou diferente. Utilizado para termos técnicos, frase idiomática, pensamento, etc...
 - ...
- Atributos
 - Action (form) – não pode ter uma url vazia
 - border (tabela) – permite "1" e valor vazio
 - Enctype (form) – suporta tipo text/plain
 - ...

Elementos obsoletos

- Elementos de apresentação (melhor com CSS)
 - basefont
 - Big
 - Center
 - Font
 - Strike
 - Tt

Elementos obsoletos

- Elementos que prejudicam usabilidade e acessibilidade
 - Frame
 - Frameset
 - Noframes
- Elementos não usados ou que sua função pode ser manipulada por outros elementos
 - Acronym → abbr
 - Applet → object
 - Isindex
 - Dir → ul

Atributos obsoletos

- rev e charset (link e a)
- target (link)
- summary (table)
- Align/valign (caption iframe, input, object, hr,h1...h6, p, col, colgroup....)
- height (td/th)
- Size (hr)

API's

- HTML introduziu novas APIs e estendeu, modificou ou tornou obsoleta algumas APIs
 - Elementos de mídia
 - Video e audio
 - Controlam a execução, sincronizando múltiplos elementos de mídia e faixas de texto (e.g. legendas)

```
<video src="http://v2v.cc/~j/theora_testsuite/320x240.ogg" controls>
```

Seu navegador não suporta o elemento `<code>video</code>.`

```
</video>
```

```
<audio src="/test/audio.ogg">
```

`<p>`Seu navegador não suporta o elemento audio.`</p>`

```
</audio>
```

API's

- Atributos
 - controls : Mostra os controles padrão para o áudio/video na página.
 - autoplay : Faz com que o áudio/video reproduza automaticamente.
 - loop : Faz com que o áudio repita automaticamente.
 - width|height

API's

- É possível controlar a execução via Javascript
 - play() – inicia a execução
 - pause() – pausa a execução
 - .seekable.start() – retorna o tempo em que o arquivo começa
 - Seekable.end() – retorna o tempo em que o arquivo termina
 - currentTime = x – avança para x segundos
 - .played.end() – retorna o número de segundos que o arquivo reproduziu

API's

- Vários arquivos podem ser especificados utilizando o elemento `<source>`
 - Formatos diferentes para diferentes suportes

`<video controls>`

`<source src="foo.ogg" type="video/ogg">`

`<source src="foo.mp4" type="video/mp4">`

Seu navegador não suporta o elemento `<code>video</code>`.

`</video>`

Exercícios

- 1) Use as páginas que você criou anteriormente (ou a própria AP1!) e transforme as divs nas novas tags semânticas
- 2) Crie uma nova página (pode usar a mesma estrutura da anterior), e adicione um vídeo e um áudio
 - Bônus: faça com que, se o áudio tocar, o vídeo pare (e vice-versa)
 - Dica: javascript, `onplaying`
- 3) Defina um div que contenha um círculo, e faça dois botões: um que aumenta o tamanho do círculo e outro que diminui.