

# Desenvolvimento de Sistemas WEB

## Prof. Msc Jacson Rodrigues

# Visão Geral do HTML

- A Web é baseada em 3 pilares:
  - Um esquema de nomes para localização de fontes de informação na Web: URI.
  - Um Protocolo de acesso para acessar estas fontes: HTTP.
  - Uma linguagem de *Hypertexto*, para a fácil navegação entre as fontes de informação: HTML.

# HTML

- Abreviação de *Hypertext Markup Language*  
Linguagem de Marcação de Hypertexto
- O HTML é uma linguagem
  - para publicação de conteúdo (texto, imagem, vídeo, áudio e etc) na Web.
- Hipertexto
  - conjuntos de elementos ligados por conexões:
    - - palavras, imagens, vídeos, áudio, documentos, etc.
- Estes elementos ligam uma HTML a outra, fornecendo uma rede enorme de informações.

# HTML

- Distribuição de uma maneira global:
  - Necessita de uma uma linguagem única;
  - Entendida universalmente por diversos meios de acesso.
- O HTML se propõe a ser esta linguagem.
- Desenvolvido originalmente por Tim Berners-Lee
- O HTML ganhou popularidade com o Mosaic:
  - navegador desenvolvido por Marc Andreessen na década de 1990.
- Desde então:
  - Utilização do HTML com as mesmas convenções.

# HTML

- Entre 1993 e 1995:
  - versões HTML+, HTML2.0 e HTML3.0;
- Em 1997:
  - Versão 3.2;
  - Realizada pela W3C, responsável por manter o padrão do código.
- Intenções do HTML:
  - Uma linguagem independente de plataformas, navegadores e outros meios de acesso;
  - Não deixar a Web ser proprietária, incompatível ou limitada.

# HTML5

- Inicialmente desenvolvida pela:
  - *Web Hypertext Application Technology Working Group* – WHATWG
- Em busca de:
  - Mais flexibilidade para as produções Web.
- A participação no grupo é livre e você pode se inscrever na lista de e-mail para contribuir.
- Código reconhecido por todos em 2006, inclusive pela W3C.
- Código com mais semântica que o HTML4.

# Objetivos do HTML5

- Facilitar a manipulação do elemento
  - Possibilita ao desenvolvedor modificar as características dos objetos;
  - De forma não intrusiva e
  - De maneira que seja transparente para o usuário.
- Fornece ferramentas para bom funcionamento de:
  - CSS e
  - Javascript.
- Cria novas tags e modifica a função de outras:
  - Seções comuns e específicas para rodapé, cabeçalho, sidebar, menus, etc.
- Traz mais semântica com menos código.

# Suporte dos navegadores

- Desenvolvimento modular:
  - HTML5 e CSS3
- Motores de Renderização:
  - Cada navegador utiliza um motor de renderização responsável pelo processamento do código da página.

Motor	Browser
Webkit	Safari, Google Chrome
Gecko	Firefox, Mozilla, Camino
Trident	Internet Explorer 4 ao 9
Presto	Opera 7 ao 10



# Suporte dos navegadores

- Técnicas de Detecção
  - suporte a geolocalização:
    - Verificar uma determinada propriedade em objetos globais: WINDOW ou NAVIGATOR.
  - Suporte de propriedades:
    - Criar um elemento e verificar se uma determinada propriedade existe neste elemento.
  - Formatos de vídeo suportados:
    - Criar um elemento e verificar se um determinado método existe neste elemento, pedindo-o de volta.
  - Suporte de tipos:
    - Criar um elemento e defina um atributo com um determinado valor, então verifique se o atributo suporta este valor.

# Suporte dos navegadores

- Utilizando o Modernizr:
  - biblioteca de detecção que lhe permite verificar o suporte da maioria das características do HTML5 e CSS3.
- Roda no *head* do documento.
- Exemplo de Geolocalização:

```
if (Modernizr.geolocation) {  
    // Aceita a feature  
} else {  
    // Não aceita a feature testada.  
}
```

# Estrutura básica

```
1 <!DOCTYPE HTML>
2 <html lang="pt-br">
3 <head>
4 <meta charset="UTF-8">
5 <link rel="stylesheet" type="text/css" href="estilo.css">
6 <title></title>
7 </head>
8 <body>
9
10 </body>
11 </html>
```

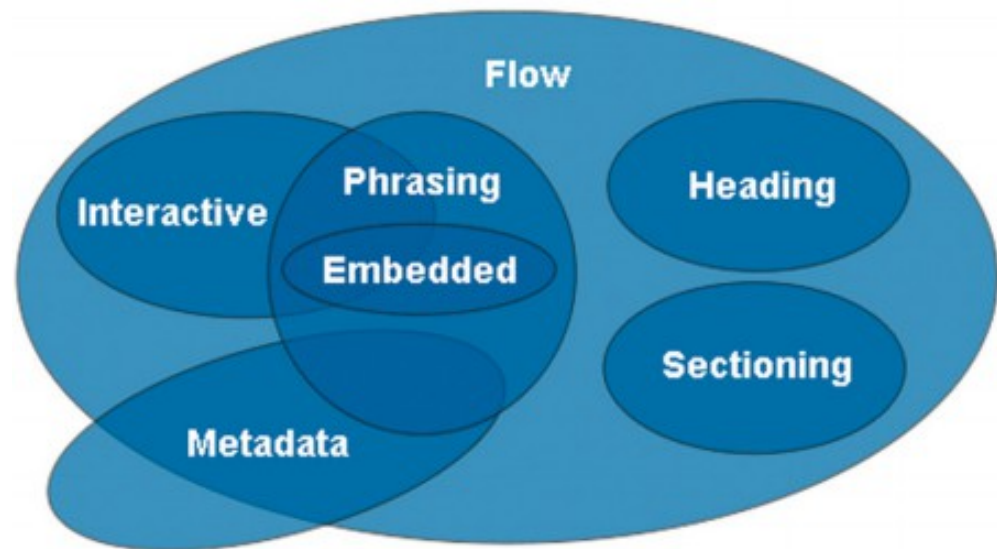
- DOCTYPE:
  - Indica a especificação de código a utilizar.
  - Não é tag html e sim uma informação para o navegador.
- HTML:
  - Indica onde começam os elementos em árvore do HTML;
- HEAD:
  - Onde fica a parte inteligente da página;
- BODY:
  - Conteúdo da página.

# Modelos de Conteúdo

- Elementos de linha
  - Marcam, na maioria das vezes, texto. Exemplos:
    - a, strong, em, img, input, abbr, span.
- Elementos de bloco
  - São como caixas, que dividem o conteúdo nas seções do layout.
- Premissas:
  - Elementos de linha podem conter outros elementos de linha
    - Depende da categoria. Ex: o elemento “a” não pode conter o elemento “label”.
  - Elementos de linha nunca podem conter elementos de bloco.
  - Elementos de bloco sempre podem conter elementos de linha.
  - Elementos de bloco podem conter elementos de bloco
    - Depende da categoria. Ex: um parágrafo não pode conter um DIV. Mas o contrário é possível.

# Categorias

- Cada elemento no HTML pode ou não fazer parte de um grupo de elementos com características similares.
  - Metadata content
  - Flow content
  - Sectioning content
  - Heading content
  - Phrasing content
  - Embedded content
  - Interactive content



# Novos elementos e atributos

- Section
  - Define uma nova seção genérica no documento. Exemplo de seções:
    - introdução ou destaque, novidades, informação de contato e chamadas para conteúdo interno.
- Nav
  - Representa uma seção da página que contém links para outras partes do website.
  - Nem todos os grupos de links devem ser elementos nav, apenas aqueles grupos que contém links importantes.
- Article
  - Representa uma parte da página que poderá ser distribuído e reutilizável em FEEDs, por exemplo.
- Aside
  - Representa um bloco de conteúdo que referencia elementos a sua volta.

# Novos elementos e atributos

- Hgroup
  - Grupo de títulos: <h1> até <h6>
- Header
  - Representa um grupo de introdução ou elementos de navegação.
  - Pode ser utilizado para agrupar índices de conteúdos, campos de busca ou até mesmo logos.
- Footer
  - Representa literalmente o rodapé da página.
- Time
  - Serve para marcar parte do texto que exhibe um horário ou uma data precisa no calendário gregoriano.

# Novos tipos de campos

- Input:
  - Tel: Telefone.
  - Search: Um campo de busca
  - Email: E-mail, com formatação e validação.
  - Url: Um endereço web, com formatação e validação.
  - Conjunto de types para datas e horas (com “step” em segundos):
    - Datetime
    - Date
    - Month
    - Week
    - Time
    - Datetime-local



# Novos tipos de campos

- Input:

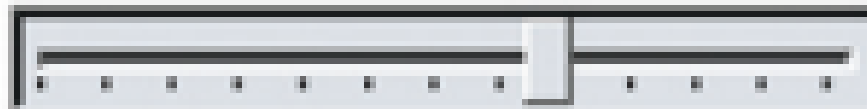
- Number.

- Exemplo:

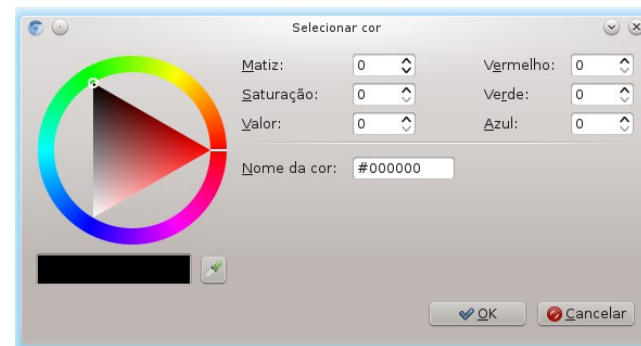
```
<input name="valuex" type="number" value="12.4" step="0.2" min="0" max="20" />
```



- Range:



- Color: Seletor de cor.



# Novos tipos de campos

- Input:
  - Autofocus.
  - Placeholder text:
  - Required:
    - Para tornar um campo de formulário obrigatório;
  - Maxlength.
  - Pattern:
    - Permite definir expressões regulares de validação sem Javascript  
`<input name="CEP" id="CEP" required pattern="\d{5}-?\d{3}" />`
  - novalidate e formnovalidate

mensagem:

# Custom Validators

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4 <meta charset="UTF-8" />
5 <title>Custom validator</title>
6 <!-- O arquivo cpf.js contém a função validaCPF, que
7 recebe uma string e retorna true ou false. -->
8 <script src="cpf.js"></script>
9 <script>
10 function vCPF(i){
11     i.setCustomValidity(validaCPF(i.value)?'' : 'CPF inválido!')
12 }
13 </script>
14 </head>
15
16 <body>
17 <form>
18 <label>CPF: <input name="cpf" oninput="vCPF(this)" /></label>
19 <input type="submit" value="Enviar" />
20 </form>
21 </body>
22
23 </html>
```

# Mais elementos

- Detalhes e Sumário

`<details>`

`<summary>Copiando <progress max="39248" value="14718"> 37,5%</summary>`

`<dl>`

`<dt>Tamanho total:</dt>`

`<dd>39.248KB</dd>`

`<dt>Transferido:</dt>`

`<dd>14.718</dd>`

`<dt>Taxa de transferência:</dt>`

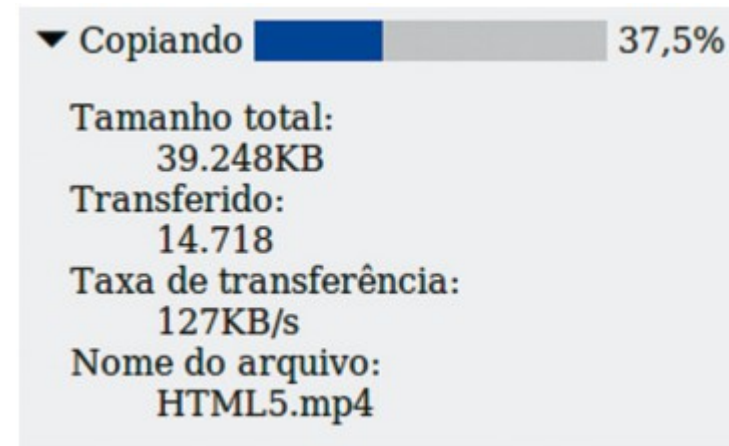
`<dd>127KB/s</dd>`

`<dt>Nome do arquivo:</dt>`

`<dd>HTML5.mp4</dd>`

`</dl>`

`</details>`



# Mais elementos

- Contenteditable
  - para tornar um elemento do HTML editável, basta incluir nele este atributo.

# Drag and Drop e Ortografia

- Drag and Drop
  - Deve-se:
    - inserir o atributo `draggable="true"` no elemento
    - Fornece os eventos:
      - Dragstart: O objeto começou a ser arrastado;
      - Drag: O objeto está sendo arrastado;
      - Dragend: A ação de arrastar terminou;
    - Recebe os eventos:
      - Dragenter: O objeto sendo arrastado entrou no objeto target;
      - Dragleave: O objeto sendo arrastado deixou o objeto target;
      - Dragover: O objeto sendo arrastado se move sobre o objeto target;
      - Drop: O objeto sendo arrastado foi solto sobre o objeto target.
- Correção Ortográfica:
  - `spellcheck="true"`

# Elementos de áudio e vídeo

- Áudio:

- ```
<audio src="mus.ogg" controls="true" autoplay="true" />
```

- OU:

- ```
<audio controls="true" autoplay="true">
```

- ```
<source src="mus.oga" />
```

- ```
<source src="mus.mp3" />
```

- ```
<source src="mus.wma" />
```

- ```
<p>Faça o <a href="mus.mp3">download da música</a>.</p>
```

- ```
</audio>
```

- Vídeo

- ```
<video src="u.ogv" width="400" height="300" />
```

# Elementos de áudio e vídeo

- Para evitar o download da mídia e informar o tipo de áudio/vídeo:

```
<source src='video.ogv'  
type='video/ogg; codecs="theora, vorbis" '>
```

```
<source src='video.mp4'  
type='video/mp4; codecs="mp4v.20.240, mp4a.40.2" '>
```



# Device

- Elemento de acesso à webcam do usuário:
  - `<device type="media">`

```
1 <!DOCTYPE html>
2 <html lang="en-US">
3 <head>
4 <meta charset="UTF-8" />
5 <title>Videochat, step 1</title>
6
7 <script>
8 function update(stream) {
9   document.getElementsByTagName('video')[0].src = stream.url;
10 }
11 </script>
12
13 </head>
14
15 <body>
16
17 <p>To start chatting, select a video camera: <device type=media
18 <video autoplay />
19
```

# Mais novidades

- MathML

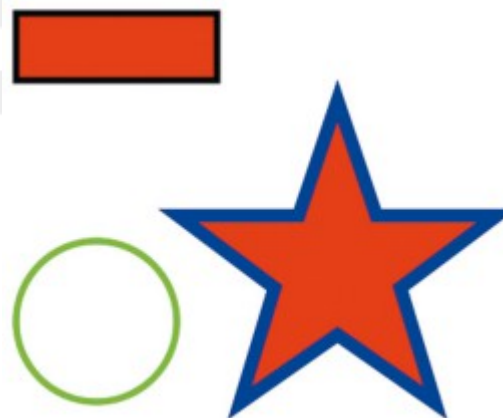
```
<math>
  <mrow>
    <mi>x</mi>
    <mo>=</mo>
    <mfrac>
      <mrow>
        <mo form="prefix">&minus;</mo>
        <mi>b</mi>
        <mo>&PlusMinus;</mo>
      </mrow>
      <msqrt>
        <msup>
          <mi>b</mi>
          <mn>2</mn>
        </msup>
        <mo>&minus;</mo>
        <mn>4</mn>
        <mo>&InvisibleTimes;</mo>
        <mi>a</mi>
        <mo>&InvisibleTimes;</mo>
        <mi>c</mi>
      </msqrt>
    </mfrac>
  </mrow>
  <mrow>
    <mn>2</mn>
    <mo>&InvisibleTimes;</mo>
    <mi>a</mi>
  </mrow>
</mfrac>
</mrow>
</math>
```

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

# Mais novidades

- SVG

```
9 <svg width="400" height="400">
10
11 <!-- Um retângulo: -->
12 <rect x="10" y="10" width="150" height="50" stroke="#000000" stroke-wi-
13 dth="5" fill="#FF0000" />
14 <!-- Um polígono: -->
15 <polygon fill="red" stroke="blue" stroke-width="10"
16 points="250,75 279,161 369,161 297,215
17 323,301 250,250 177,301 203,215
18 131,161 221,161" />
19
20 <!-- Um círculo -->
21 <circle cx="70" cy="240" r="60" stroke="#00FF00" stroke-width="5"
22 fill="#FFFFFF" />
23 </svg>
```



# Canvas API

- A Canvas API permite desenhar na tela do navegador via Javascript.
  - Elemento HTML: canvas
  - O resto todo é feito via Javascript.

```
<canvas id="x" width="300" height="300"></canvas>
```
  - Para desenhar no canvas, deve-se obter o local:

```
context=document.getElementById('x').getContext('2d')
```
  - E desenhar:

```
context.fillRect(10, 10, 50, 150); //Um retângulo
```