APRENDA DO ZERO E DO JEITO CERTO

BREVE HISTÓRICO DA INTERNET

1957 - Surge a ARPA
1969 - Primeiro e-mail é transmitido em rede
1969 - A ARPA NET começa a funcionar
197* - A ARPA NET em Universidade
1971 - Vinton Cerf cunha o termo internet
1973 - Primeira conexão entre continentes
1977 - Criação do TCP/IP
1979 - Surge a USENET
1981 - SURGE o IBM PC
1983 - Criada a MILNET
1983 - Adotado o TCP/IP
1984 - A ARPA NET possui 1000 servidores
(DNS, BBS, IRC, USENET, E-MAIL, TCP/IP...)

BREVE HISTÓRICO DA WEB

Motivação
Linguagem HTML
Protocolo HTTP
URI

1980 - A TBL cria o ENQUIRE
1989 - A TBL e Robert Cailliau fizeram a primeira comunicação
bem sucedida entre um cliente HTTP e um Servidor na Internet
1990 - Surge então o HTML
1993 - Primeiro Servidor HTTP
1993 - Primeira Página da Web

BREVE HISTÓRICO DO HTML

O HTML é uma linguagem de marcação utilizada para desenvolvimento de sites.
Esta linguagem surgiu junto com o HTTP, ambos possibilitaram a internet ser popularizada.

O HTML foi criado em 1991, por Tim Berners-Lee, no CERN
(European Council for Nuclear Research) na suíça. Inicialmente o HTML foi projetado
para interligar instituições de pesquisa próximas, e compartilhar documentos
com facilidade.

Em 1992, foi liberada a biblioteca de desenvolvimento WWW (World Wide Web),
uma rede de alcance mundial, que junto com o HTML proporcionou o uso em
escala mundial da WEB.

LINGUAGEM DE MARCAÇÃO

Ex:

Olá, meu nome é
<nome>Luiz</nome>, sou filho de
<nome>Ana</nome> e moro em
<cidade>São Paulo - SP</cidade>

R:
Olá, meu nome é Luiz, sou filho de Ana e moro em São Paulo - SP

Veremos exemplos real no decorrer do ebook.

HIPERTEXTO

Termo cunhado por Ted Nelson em 1965

Hipertexto é o termo que remete a um texto ao qual se agregam outros conjuntos de informação na forma de blocos de textos, palavras, imagens ou sons, cujo acesso se dá através de referências específicas, no meio digital denominadas hiperlinks, ou simplesmente links. Esses links ocorrem na forma de termos destacados no corpo de texto principal, ícones gráficos ou imagens e têm a função de interconectar os diversos conjuntos de informação, oferecendo acesso sob demanda às informações que estendem ou complementam o texto principal. O conceito de "linkar" ou de "ligar" textos foi criado por Ted Nelson nos anos 1960 e teve como influência o pensador francês Roland Barthes, que concebeu em seu livro S/Z o conceito de "Lexia", que seria a ligação de textos com outros textos. Em termos mais simples, o hipertexto é uma ligação que facilita a navegação dos internautas. Um texto pode ter diversas palavras, imagens ou até mesmo sons que, ao serem clicados, são remetidos para outra página onde se esclarece com mais precisão o assunto do link abordado.

O sistema de hipertexto mais conhecido atualmente é a World Wide Web, no entanto a Internet não é o único suporte onde este modelo de organização da informação e produção textual se manifesta.

Derivada da SGML e HyTime

HIPERMÍDIA

Autoria atribuída a Ted Nelson é o hipertexto que não está limitado a conter apenas textos, podendo possuir imagens, vídeos, sons e etc.

O QUE SÃO TAGS?

Tags são o conjunto de caracteres que formam um elemento, ou seja, quando nos referenciamos à Tag "p" estamos falando disso: <p>

Existem dois tipos de Tags, as que necessitam de fechamento e as que não necessitam de fechamento.

Para as que necessitam de fechamento, utilizamos o sinal de menor (<), seguido do nome do elemento e o sinal de maior (>) para abertura. Para fechamento, utilizamos o sinal de menor (<), seguido de barra (/), nome do elemento e o sinal de maior (>).

O QUE SÃO TAGS?

```
<!-- Exemplo de elemento que necessita de fechamento -->
<p>A tag do elemento de parágrafo necessita de fechamento.</p>
```

Os elementos que não necessitam de fechamento, também conhecidos como elementos vazios, somente utilizamos o sinal de menor (<), seguido do nome do elemento e o sinal de maior (>).

```
<!-- Exemplo de elemento vazio -->
Texto utilizando<br>
quebra de linha
```

O QUE SÃO ELEMENTOS?

Elementos são formados a partir de Tags e entre as Tags é que está o conteúdo do Elemento. Se quisermos exibir um parágrafo em um site utilizamos a Tag <p> que semanticamente quer dizer que o conteúdo que se espera nesse Elemento é um parágrafo.

Alguns exemplos:

```
<!-- A Tag <label> define que o conteúdo do Elemento é uma <label> (rótulo)  -->
<label>Informe o seu nome</label>

<!-- A Tag <address> define que o conteúdo do Elemento é um endereço (endereços físicos à virtuais)  -->
<address>
    Aprenda a Utilizar Todo o Poder do HTML5
    <a href="mailto:contato@htmlecsspro.com">Contato</a>
</address>
```

O QUE SÃO ELEMENTOS?

Não confunda Tags com Elementos

As Tags servem para marcar onde começam e terminam os Elementos. Já os Elementos são uma parte conceitual/semântica que têm um começo e fim determinados.

Parece uma diferença boba, mas mantenha ela sempre em mente e isso vai fazer toda a diferença no seu entendimento de **HTML**.

O QUE SÃO ATRIBUTOS?

Atributos são informações que passamos na Tag para que ela se comporte da maneira esperada. Existem atributos globais (que funcionam em todas as Tags) e específicos (que são direcionados para cada Tag, através de especificação).

Os Atributos possuem nome e um valor, existem Atributos que você vai usar praticamente sempre e existem outros que serão mais raros.

O QUE SÃO ATRIBUTOS?

Atributos Globais

Seguem alguns atributos globais importantes e suas descrições simplificadas:

Accesskey Determina uma (ou mais) tecla(s) de atalho para o elemento. Para definir mais de uma tecla, coloque-as separadas por espaço.
Class Determina uma (ou mais) classe(s) para o elemento. Para definir mais de uma classe, coloque-as separadas por espaço.
Draggable Define se um elemento é arrastável ou não.
Hidden Oculta o elemento onde for declarado.
ID É o identificador único do elemento. Somente deve ser declarado um id por elemento. E este id não deve ser repetido na mesma página.
Lang Determina o idioma em que está escrito o conteúdo do elemento.
Style Determina propriedades CSS diretamente no elemento.
Tabindex Organiza os elementos por ordem de tabulação. Deve-se usar valor numérico.
Title Representa um auxílio ao elemento. Semelhante a dica do elemento.

Na prática
Agora que você entendeu separadamente o papel das Tags, Elementos e Atributos, vamos ver um exemplo prático:

```
<!-- A Tag <img> define que o conteúdo do Elemento é uma imagem e os atributos que utilizamos são src e alt -->
<img src="imagem.jpg" alt="HTML e CSS Pro" />
```

O QUE É SEMÂNTICA NA WEB?

A Web Semântica é nada mais nada menos, que uma web com toda sua informação organizada de forma que não somente seres humanos possam entendê-la, mas principalmente máquinas. As máquinas nos ajudarão em tarefas que hoje fazemos manualmente.

O QUE É WEB STANDARDS?

São normas criadas pelo **W3C**, a serem implementadas pelos Browsers do mercado e seguidas pela comunidade de desenvolvimento.

AS TAGS DO HTML (até 4.01)

AS TAGS DO HTML

<html>
Representa um elemento HTML

<html> - atributo lang
<html lang="pr-br">

Usado para que os **user-agents** saibam
qual é a linguagem principal do documento

AS TAGS DO HTML

<head>
Contém meta-dados a respeito do documento

<title>
Representa o título da página, que constará na barra de título do navegador.

Ex:

```
<title>
    Título da página que aparecerá no navegador
</title>
```

AS TAGS DO HTML

<body>
Representa o corpo da página

Ex:

```
<body>
    Aqui vai o conteúdo o corpo da página do site, é aqui que vamos combinando
    as tags e formando nossa página de acordo com a nossa necessidade.
</body>
```

AS TAGS DO HTML

<h1> a **<h6>** (Headings = Títulos)
Representam o cabeçalho de diferentes níveis de importância do documento

Ex:

```
<h1>Título 1</h1>
<h2>Título 2</h2>
<h3>Título 3</h3>
<h4>Título 4</h4>
<h5>Título 5</h5>
<h6>Título 6</h6>
```

Por padrão o H1 tem a fonte maior que o H2 e assim sucessivamente

AS TAGS DO HTML

<a> - atributo href

Representa uma localização para um trecho específico desse recurso

Ex:

```
<a href="index.html">Home do site</a>
<a href="mailto:contato@htmlecsspro.com">contato@htmlecsspro.com</a>
<a href="http://google.com">Google</a>
```

AS TAGS DO HTML

<a> - atributo name

Nomeia um ponto específico de um documento para
que possa ser referenciado por outros links

Ex:

```
<a name="topo"></a>
```

Ponto para onde será direcionado ao clicar em Topo

OBS: caso este código esteja no meio da página, ao clicar em Topo, será direcionado para a posição onde o mesmo se encontra

```
<a href="#topo">Topo</a>
```

Link que será clicado

AS TAGS DO HTML

<img>
Representa uma imagem

<img> - atributo src
Representa o caminho (absoluto ou relativo) da imagem

Ex:

```
<img src="imagem.jpg" alt="HTML e CSS Pro" />
```

AS TAGS DO HTML

Representa um paragrafo

Ex:

Aqui vai o texto do paragrafo da página.
Pode ter vários parágrafos numa única página.

AS TAGS DO HTML

<table>
Tag usada para apresentar dados em formato tabular

<tr>
Representa uma linha de uma tabela

<td>
Representa uma célula de uma tabela

<td> - atributo colspan
Números de colunas na qual a célula irá se expandir

<td> - atributo rowspan
Números de linhas na qual a célula irá se expandir

AS TAGS DO HTML

<table>, <tr> e <td>

Ex:

```
<table>
    <tr>
        <td>Nome do produto</td>
        <td>Valor do produto</td>
    </tr>
    <tr>
        <td>Caderno capa dura</td>
        <td>R$ 50,00</td>
    </tr>
</table>
```

AS TAGS DO HTML

<table>, <td rowspan>

Ex:

```
<table>
	<tr>
		<th>Nome produto</th>
		<th>Preço</th>
		<th>Total</th>
	</tr>
	<tr>
		<td>Camisa polo</td>
		<td>R$ 50</td>
		<td rowspan="2">R$ 200</td>
	</tr>
	<tr>
		<td>Calça Jeans</td>
		<td>R$ 150</td>
	</tr>
</table>
```

RESULTADO

Nome produto	Preço	Total
Camisa polo	R$ 50,00	R$ 200,00
Calça Jeans	R$ 150,00	

Ocupamos duas linhas com atributo ROWSPAN na segunda linha da terceira coluna

AS TAGS DO HTML

<table>, <td colspan>

Ex:

```
<table>
	<tr>
		<th>Nome produto</th>
		<th>Preço</th>
	</tr>
	<tr>
		<td>Camisa polo</td>
		<td>R$ 50</td>
	</tr>
	<tr>
		<td colspan="2" align="center">R$ 50</td>
	</tr>
</table>
```

RESULTADO

Nome produto	Preço
Camisa polo	R$ 50,00
R$ 50,00	

Ocupamos duas colunas horizontalmente na terceira linha com atributo COLSPAN

AS TAGS DO HTML

<form> - atributos

<form> </form> representa um formulário

Action
<form action="pagina.html"></form>

Representa a URL do redirecionamento que irá receber os dados do formulário

AS TAGS DO HTML

<form> - atributos

Method GET

<form action="pagina.html" method="get"></form>

Representa o método HTTP que será usado para o envio do formulário

OBS: O método GET submete o formulário com os valores dos campos concatenados à URL

Ex:

http://www.site.com.br/dados?nome=luiz&sobrenome=vieira

AS TAGS DO HTML

<form> - atributos

Method POST (default)

```
<form action="pagina.html" method="post"></form>
```

Os dados do formulário são enviados nos headers das solicitações HTTP

AS TAGS DO HTML

<input>
Representa um campo de um formulário

<input> - atributos

type
<text>, <button>, <checkbox>, <radio>, <file>, <hidden>, <image>,
<password>, <reset>, <submit>

AS TAGS DO HTML

<select>
Representa uma caixa de seleção

<option>
Representa uma opção de uma seleção

<option> - atributo value
O valor da opção. É o que será submetido ao servidor

AS TAGS DO HTML

<textarea> - atributo rows

Número de linhas

<textarea> - atributo cols

Número de colunas

AS TAGS DO HTML

<button>
Representa um botão

<label>
Representa um rótulo

<label> - atributo FOR
Espedifica o ID do campo ao qual a label está associada

AS TAGS DO HTML

 e
Representam uma divisão em um site

<hr>
Representa uma linha horizontal

 e
Representam uma lista ordenada (ol) e uma lista não ordenada (ul)

Representa um item de lista

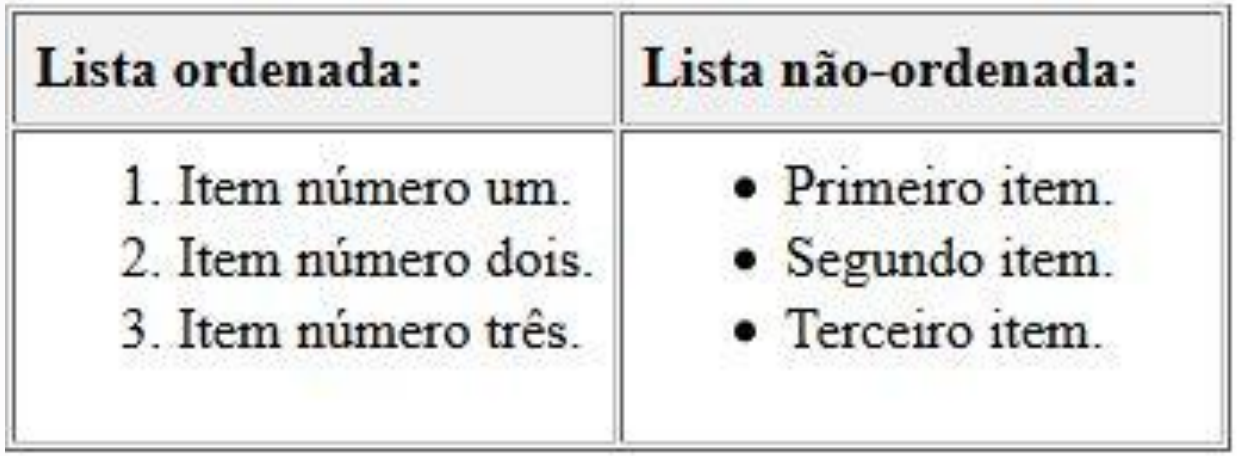
AS TAGS DO HTML

```
<ol>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 2</li>
</ol>
```

Lista ordenada(ol)

```
<ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 2</li>
</ul>
```

Lista não ordenada(ul)



Lista ordenada:	Lista não-ordenada:
1. Item número um. 2. Item número dois. 3. Item número três.	• Primeiro item. • Segundo item. • Terceiro item.

AS TAGS DO HTML

Dá ênfase a algo

<iframe>
Representa uma página embutida em outra. [inline frame]

<iframe> - atributo src
Localização da página embutida

<link>
Define um relacionamento entre um documento e um recurso externo.
Suporte real para folhas de estilo e favicons

<link> - atributo href
Especifica a localização do recurso externo

<link> - atributo rel
Especifica o tipo de relacionamento

AS TAGS DO HTML

<script>
Código em linguagem de script (por padrão JavaScript), ou URI para arquivo de script externo

<script> - atributo src
Especifica a localização do script

<meta>
Representa um meta-dado sobre um documento

<meta> - atributo name
Nome do meta-dado. Velores possíveis (author, description, keywords, replay-to)

<meta> - atributo content
Valor do meta-dado

AS TAGS DO HTML

<pre>
Texto pré-formatado.
Preserva espaços e quebras de linha e é
exibido em fonte mono-espaçada

<style> - atributo src
Trecho de código CSS

AS TAGS DO HTML SEMÂNTICA

<abbr>
Define siglas e abreviações.

<address>
Define um endereço

<blockquote>
Define uma citação

<caption>
Legenda para tabelas. Por padrão é centralizada

<code>
Trecho de código

AS TAGS DO HTML SEMÂNTICA

```
<code>

</code>Código Inline</code>

<pre>
    </code>
        Código
        multilinha
    </code>
</pre>
```

AS TAGS DO HTML SEMÂNTICA

<del>
Deleta um texto de um documento

<ins>
Insere um texto no documento. Geralmente sublinhado

<fieldset>
Agrupa campos relacionados de um formulário

<legend>
Adiciona uma legenda a um fieldset

<samp>
Saída de código

AS TAGS DO HTML SEMÂNTICA

<tfoot>
Rodapé de uma tabela

<thead>
Cabeçalho de uma tabela

<th>
Célula-título de uma tabela

AS TAGS DO HTML – USE COM MODERAÇÃO

Quebra uma linha. Na dúvida não use

<b>
Deixa o texto em negrito mas não da ênfase. Prefira

<i>
Deixa o texto em itálic, prefira .

DOCTYPE

<!DOCTYPE html>

Define a versão do **html** na qual o documento foi escrito,
Fazendo com que o navegador renderize uma
página corretamente

CSS

LINGUAGEM CSS

Separação de responsabilidades
Folha de estilo em cascata
Compartibilidade de crossbrowser
Propriedades
Regras
Seletores
Pseudo-elementos
Pseudo-classes

SELETORES DO CSS (ATÉ O 2)

*[Universal]
E > F[filhos]
E:first-child [pseudo-classe first-chid]
E:link
E:visited[pseudo-classe de link]
E:active, E:hover,
E:focus[pseudo-classe dinâmicas]
E + F[adjacenter]
E[foo][possui atributo]
E[foo="warning"][possui atributo igual a]
E[foo~="warning"][possui atributo independente de possuir espaços]
div.classe
E#id[id]

COMENTÁRIOS EM CSS (ATÉ O 2)

```
/*Este é um comentários*/
```

PROPRIEDADES DO CSS (ATÉ O 2)

Background
Background-color
Background-image
Background-repeat
...
Border
Height
Width
Max-height
Max-width
Min-height
Min-width
Font-Family
Font-size
Font-style
Font-weight
List-style-type (square, circle, bullet, none)

Margin
Padding
Cursor
Display (none, block, inline-block, ...)
Overflow (visible, hidden, scroll, auto)
Position
Bottom
Right
Left
Top
Z-index
Color
Letter-spacing
Line-height
Text-align
Text-decoration (underline, overline, line-through, none)
Text-indent

PROPRIEDADES DO CSS (ATÉ O 2)

Text-transform (uppercase, capitalize, lowercase)
Vertical-align
Word-spacing
Float

HTML

VISÃO GERAL DO HTML5

De acordo com o **W3C** a Web é baseada em 3 pilares:

• Um esquema de nomes para localização de fontes de informação na Web, esse esquema chama-se **URI**.

• Um Protocolo de acesso para acessar estas fontes, hoje o **HTTP**.

• Uma linguagem de **Hypertexto**, para a fácil navegação entre as fontes de informação: o **HTML**.

VISÃO GERAL DO HTML5

O começo e a interoperabilidade

Entre 1993 e 1995, o **HTML** ganhou as versões **HTML+**, **HTML2.0** e **HTML3.0**, onde foram propostas diversas mudanças para enriquecer as possibilidades da linguagem. Contudo, até aqui o **HTML** ainda não era tratado como um padrão. Apenas em 1997, o grupo de trabalho do **W3C** responsável por manter o padrão do código, trabalhou na versão 3.2 da linguagem, fazendo com que ela fosse tratada como prática comum.
Você pode ver: http://www.w3.org/TR/html401/appendix/changes.html.

Desde o começo o **HTML** foi criado para ser uma linguagem independente de plataformas, browsers e outros meios de acesso. Interoperabilidade significa menos custo. Você cria apenas um código **HTML** e este código pode ser lido por diversos meios, ao invés de versões diferentes para diversos dispositivos. Dessa forma, evitou-se que a Web fosse desenvolvida em uma base proprietária, com formatos incompatíveis e limitada.

Por isso o **HTML** foi desenvolvido para que essa barreira fosse ultrapassada, fazendo com que a informação publicada por meio deste código fosse acessível por dispositivos e outros meios com características diferentes, não importando o tamanho da tela, resolução, variação de cor. Dispositivos próprios para deficientes visuais e auditivos ou dispositivos móveis e portáteis. O **HTML** deve ser entendido universalmente, dando a possibilidade para a reutilização dessa informação de acordo com as limitações de cada meio de acesso.

VISÃO GERAL DO HTML5

WHAT Working Group

Enquanto o **W3C** focava suas atenções para a criação da segunda versão do **XHTML**, um grupo chamado Web Hypertext Application Technology **Working Group** ou **WHATWG** trabalhava em uma versão do **HTML** que trazia mais flexibilidade para a produção de websites e sistemas baseados na web.

O **WHATWG** (http://www.whatwg.org/) foi fundado por desenvolvedores de empresas como Mozilla, Apple e Opera em 2004. Eles não estavam felizes com o caminho que a Web tomava e nem com o rumo dado ao **XHTML**. Por isso, estas organizações se juntaram para escrever o que seria chamado hoje de **HTML5**.

Entre outros assuntos que o **WHATWG** se focava era Web Forms 2.0 que foi incluído no **HTML5** e o **Web Controls 1.0** que foi abandonado por enquanto.

A participação no grupo é livre e você pode se inscrever na lista de email para contribuir.

Por volta de 2006, o trabalho do **WHATWG** passou ser conhecido pelo mundo e principalmente pelo **W3C** - que até então trabalhavam separadamente - que reconheceu todo o trabalho do grupo. Em Outubro de 2006, Tim Berners-Lee anunciou que trabalharia juntamente com o **WHATWG** na produção do **HTML5** em detrimento do **XHTML 2**. Contudo o **XHTML** continuaria sendo mantido paralelamente de acordo comas mudanças causadas no **HTML**. O grupo que estava cuidando especificamente do **XHTML 2** foi descontinuado em 2009.

VISÃO GERAL DO HTML5

O HTML5 e suas mudanças

Quando o **HTML4** foi lançado, o **W3C** alertou os desenvolvedores sobre algumas boas práticas que deveriam ser seguidas ao produzir códigos client-side. Desde este tempo, assuntos como a separação da estrutura do código com a formatação e princípios de acessibilidade foram trazidos para discussões e à atenção dos fabricantes e desenvolvedores.

Contudo, o **HTML4** ainda não trazia diferencial real para a semântica do código. o **HTML4** também não facilitava a manipulação dos elementos via **Javascript** ou **CSS**. Se você quisesse criar um sistema com a possibilidade de Drag'n Drop de elementos, era necessário criar um grande script, com bugs e que muitas vezes não funcionavam de acordo em todos os browsers.

VISÃO GERAL DO HTML5

O que é o HTML5?

O **HTML5** é a nova versão do **HTML4**. Enquanto o **WHATWG** define as regras de marcação que usaremos no **HTML5** e no **XHTML**, eles também definem **APIs** que formarão a base da arquitetura web. Essas **APIs** são conhecidas como **DOM Level 0**.

Um dos principais objetivos do **HTML5** é facilitar a manipulação do elemento possibilitando o desenvolvedor a modificar as características dos objetos de forma não intrusiva e de maneira que seja transparente para o usuário final.

Ao contrário das versões anteriores, o **HTML5** fornece ferramentas para a **CSS** e o **Javascript** fazerem seu trabalho da melhor maneira possível. O **HTML5** permite por meio de suas **APIs** a manipulação das características destes elementos, de forma que o website ou a aplicação continue leve e funcional.

O **HTML5** também cria novas tags e modifica a função de outras. As versões antigas do **HTML** não continham um padrão universal para a criação de seções comuns e específicas como **rodapé**, **cabeçalho**, **sidebar**, **menus** e etc. Não havia um padrão de nomenclatura de **IDs**, **Classes** ou **tags**. Não havia um método de capturar de maneira automática as informações localizadas nos rodapés dos websites.

Há outros elementos e atributos que sua função e significado foram modificados e que agora podem ser reutilizados de forma mais eficaz. Por exemplo, elementos como **B** ou **I** que foram descontinuados em versões anteriores do **HTML** agora assumem funções diferentes e entregam mais significado para os usuários.

O **HTML5** modifica a forma de como escrevemos código e organizamos a informação na página. Seria mais semântica com menos código. Seria mais interatividade sem a necessidade de instalação de plugins e perda de performance. É a criação de código interoperável, pronto para futuros dispositivos e que facilita a reutilização da informação de diversas formas.

O **WHATWG** tem mantido o foco para manter a retrocompatibilidade.

Nenhum site deverá ter de ser refeito totalmente para se adequar aos novos conceitos e regras. O **HTML5** está sendo criado para que seja compatível com os browsers recentes, possibilitando a utilização das novas características imediatamente.

VISÃO GERAL DO HTML5

Padronização das regras de marcação

- Novas tags e alteração no comportamento de tags existentes
- Padronização de seções comuns: cabeçalho, rodapé, menu, etc.
- Padronização de nomenclaturas: melhoria na semântica
- Elimina a necessidade de scripts externos (plugins / bibliotecas)
- Manipulação não-intrusiva de objetos
- Independente de sistema operacional ou navegador
- Retrocompatível: sem necessidade de reescrita de sites antigos
- Definição de APIs base para arquitetura web (DOM Level 0)

VISÃO GERAL DO HTML5

Orientação à Semântica

Ao ler um livro, conseguimos diferenciar **parágrafos**, **títulos**, **rodapés** e **cabeçalhos** devido a formatação visual

Robôs de busca não conseguem notar essas diferenças

Cabe ao desenvolvedor marcar essas informações com tags que atribuam significado a cada seção do código

- Tags **HTML**, como <header>, <footer> e <time> foram introduzidas para auxiliar na marcação semântica do código
- Tags <b> e <i> foram mantidas, mas ganharam **semântica**
- Tags relacionadas apenas a formatação visual foram removidas já que são facilmente supridas por folhas de estilo **CSS**

VISÃO GERAL DO HTML5

Desenvolvimento modular

- Em versões antigas do **HTML** e **CSS**:
    - Todas as ideias de uma nova versão eram especificadas
    - Depois de testadas, as especificações eram então divulgadas para implementação nos navegadores

- **HTML5** e **CSS3**:
    - Desenvolvimento modular
    - Grupos diferentes cuidam de tecnologias diferentes, que são publicadas (e implementadas) de maneira independente
    - Ponto negativo: problemas de compatibilidade

VISÃO GERAL DO HTML5

Motores de renderização

- Múltiplos navegadores, cada um com suas características
- Impossível garantir 100% de compatibilidade entre browsers
- Desenvolvedores focam em manter o código compatível com os motores que processam e renderizam o código-fonte

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

VISÃO GERAL DO HTML5

Compatibilidade com HTML5

	Safari	Chrome	Opera	Firefox	IE 8	IE 9
Local Storage						
Histórico de Sessão						
Aplicações Offline						
Novos tipos de campos						
Form: Autofocus						
Form: Autocomplete						
Form: Required						
Video, Audio, Canvas Text						

VISÃO GERAL DO HTML5

Utilizando o Modernizr

O Modernizr (http://www.modernizr.com/) é uma biblioteca de detectação que lhe permite verificar o suporte da maioria das características do **HTML5** e **CSS3**. O Modernizr roda automaticamente assim que você o adiciona no head do documento.

Assim, se você quiser verificar se o browser suporta Geolocalização, por exemplo, basta inserir este script na página.
Se o browser suportar a feature testada, ele retornará true:

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

ESTRUTURA BÁSICA, DOCTYPE E CHARSETS

A estrutura básica do **HTML5** continua sendo a mesma das versões anteriores da linguagem, há apenas uma excessão na escrita do **Doctype**. Segue abaixo como a estrutura básica pode ser seguida:

```
<!DOCTYPE HTML>
<html lang="pt-br">
    <head>
        <meta charset="UTF-8">
        <link rel="stylesheet" type="text/css" href="estilo.css">
        <title>Titulo da página</title>
    </head>
    <body>
        Conteúdo
    </body>
</html>
```

ESTRUTURA BÁSICA, DOCTYPE E CHARSETS

O doctype

O Doctype deve ser a primeira linha de código do documento antes da tag **HTML**.

```
<!DOCTYPE html!>
```

O **Doctype** indica para o navegador e para outros meios qual a especificação de código utilizar. Em versões anteriores, era necessário referenciar o DTD diretamente no código do **Doctype**. Com o **HTML5**, a referência por qual DTD utilizar é responsabilidade do Browser.

O **Doctype** não é uma tag do **HTML**, mas uma instrução para que o browser tenha informações sobre qual versão de código a marcação foi escrita.

ESTRUTURA BÁSICA, DOCTYPE E CHARSETS

O elemento HTML

O código **HTML** é uma série de elementos em árvore onde alguns elementos são filhos de outros e assim por diante.
O elemento principal dessa grande árvore é sempre a tag **HTML**.

```
<html lang="pt-br">
```

O atributo **LANG** é necessário para que os **user-agents** saibam qual a linguagem principal do documento.
Lembre-se que o atributo **LANG** não é restrito ao elemento **HTML**, ele pode ser utilizado em qualquer outro elemento para indicar o idioma do texto representado.

Para encontrar a listagem de códigos das linguagens, acesse:
http://www.w3.org/International/questions/qa-choosing-language-tags.

ESTRUTURA BÁSICA, DOCTYPE E CHARSETS

HEAD

A Tag **HEAD** é onde fica toda a parte inteligente da página. No **HEAD** ficam os metadados. Metadados são informações sobre a página e o conteúdo ali publicado.

ESTRUTURA BÁSICA, DOCTYPE E CHARSETS

Metatag Charset

No nosso exemplo há uma metatag responsável por chavear qual tabela de caractéres a página está utilizando.

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

Nas versões anteriores ao **HTML5**, essa tag era escrita da forma abaixo:

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

Essa forma antiga será também suportada no **HTML5**. Contudo, é melhor que você utilize a nova forma.
A Web é acessada por pessoas do mundo inteiro. Ter um sistema ou um site que limite o acesso e pessoas de outros países é algo que vai contra a tradição e os ideais da internet. Por isso, foi criado uma tabela que suprisse essas necessidades, essa tabela se chama Unicode.

A tabela Unicode suporta algo em torno de um milhão de caracteres. Ao invés de cada região ter sua tabela de caracteres, é muito mais sensato haver uma tabela padrão com o maior número de caracteres possível. Atualmente a maioria dos sistemas e browsers utilizados por usuários suportam plenamente Unicode. Por isso, fazendo seu sistema Unicode você garante que ele será bem visualizado aqui, na China ou em qualquer outro lugar do mundo. O que o Unicode faz é fornecer um único número para cada caractere, não importa a plataforma, nem o programa, nem a língua.

ESTRUTURA BÁSICA, DOCTYPE E CHARSETS

Tag LINK

Há dois tipos de links no **HTML**: a tag A, que são links que levam o usuário para outros documentos e a tag **LINK**, que são links para fontes externas que serão usadas no documento.

No nosso exemplo há uma tag **LINK** que importa o **CSS** para nossa página:

```
<link rel="stylesheet" type="text/css" href="estilo.css">
```

O atributo rel="stylesheet" indica que aquele link é relativo a importação de um arquivo referente a folhas de estilo.

Há outros valores para o atributo **REL**, como por exemplo o **ALTERNATE**:

```
<link rel="alternate" type="application/atom+xml" title="feed" href="/feed/">
```

Neste caso, indicamos aos **user-agents** que o conteúdo do site poder ser encontrado em um caminho alternativo via Atom **FEED**.

No **HTML5** há outros links relativos que você pode inserir como o rel="archives" que indica uma referência a uma coleção de material histórico da página. Por exemplo, a página de histórico de um blog pode ser referenciada nesta tag.

MODELOS DE CONTEÚDO

Há pequenas regras básicas que nós já conhecemos e que estão no **HTML** desde o início. Estas regras definem onde os elementos podem ou não estar. Se eles podem ser filhos ou pais de outros elementos e quais os seus comportamentos.
Dentre todas as categorias de modelos de conteúdo, existem dois tipos de elementos: elementos de linha e de bloco.
Os elementos de linha marcam, na sua maioria das vezes, texto.

Alguns exemplos: <a>, <Strong>, , <img>, <input>, <abbr>,

Os elementos de blocos são como caixas, que dividem o conteúdo nas seções do layout.
Abaixo segue algumas premissas que você precisa relembrar e conhecer:
• Os elementos de linha podem conter outros elementos de linha, dependendo da categoria que ele se encontra.

Por exemplo: o elemento a não pode conter o elemento <label>.
• Os 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, dependendo da categoria que ele se encontra.

Por exemplo, um parágrafo não pode conter um . Mas o contrário é possível.
Estes dois grandes grupos podem ser divididos em categorias. Estas categorias dizem qual modelo de conteúdo o elemento trabalha e como pode ser seu comportamento.

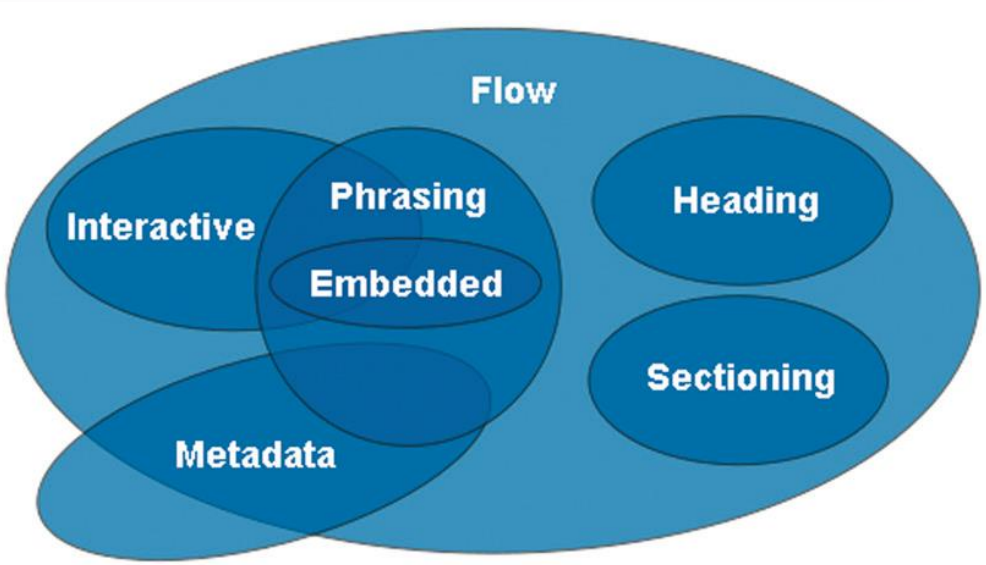
MODELOS DE CONTEÚDO

Categorias

Cada elemento no **HTML** pode ou não fazer parte de um grupo de elementos com características similares.
As categorias estão a seguir. Manteremos os nomes das categorias em inglês para que haja um melhor entendimento:

- Metadata content
- Flow content
- Sectioning content
- Heading content
- Phrasing content
- Embedded content
- Interactive contente

Abaixo segue como as categorias estão relacionadas de acordo com o **WHATWG**:



MODELOS DE CONTEÚDO

Metadata content

- Elementos que modificam a apresentação ou comportamento do resto do documento
- Elementos que fazem ligações com outros documentos

<base>, <command>, <link>, <meta>, <noscript>, <script>, <style>, <title>

Este conteúdo vem antes da apresentação, formando uma relação com o documento e seu conteúdo com outros documentos que distribuem informação por outros meios.

MODELOS DE CONTEÚDO

Flow content

Elementos que tipicamente contém texto ou elementos da categoria de conteúdo Embedded

<a>, <abbr>, <address>, <article>, <aside>, <audio>, <b>, <bdo>, <blockquote>,
, <button>, <canvas>, <cite>, <code>, <command>, <datalist>, <del>, <details>, <dfn>, , <dl>, , <embed>, <fieldset>, <figure>, <footer>, <form>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <header>, <hgroup>, <hr>, <i>, <iframe>, <img>, <input>, <ins>, <kbd>, <keygen>, <label>, <map>, <mark>, <math>, <menu>, <meter>, <nav>, <noscript>, <object>, <ol>, <output>, <p>, <pre>, <progress>, <q>, <ruby>, <samp>, <script>, <section>, <select>, <small>, , , <sub>, <sup>, <svg>, <table>, <textarea>, <time>, <ul>, <var>, <video>, <wbr>

Por via de regra, elementos que seu modelo de conteúdo permitem inserir qualquer elemento que se encaixa no Flow Content, devem ter pelo menos um descendente de texto ou um elemento descendente que faça parte da categoria embedded.

MODELOS DE CONTEÚDO

Sectioning content

Elementos usados para definir o conteúdo de uma subseção dentro de um documento

- <article> delimita publicações em um blog, notícias, etc.
- <aside> delimita dados relacionados ao conteúdo ao redor
- <nav> é usada para delimitar navegação dentro da página
- <section> é uma delimitação genérica, sem tanta semântica

<article>, <aside>, <nav>, <section>

Heading content

- Elementos que definem cabeçalhos. Normalmente presentes dentro da categoria Sectioning

<h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <hgroup>

MODELOS DE CONTEÚDO

Phrasing content

Elementos usados para definir, principalmente, texto e suas marcações, como formatação e outros atributos

<abbr>, <audio>, <b>, <bdo>,
, <button>, <canvas>, <cite>, <code>, <command>, <datalist>, <dfn>, , <embed>, <i>, <iframe>, <img>, <input>, <kbd>, <keygen>, <label>, <mark>, <math>, <meter>, <noscript>, <object>, <output>, <progress>, <q>, <ruby>, <samp>, <script>, <select>, <small>, , , <sub>, <sup>, <svg>, <textarea>, <time>, <var>, <video>, <wbr>

Embedded content

Elementos que importam informações de recursos externos ou de outras linguagens de marcação para o documento

<audio>, <canvas>, <embed>, <iframe>, <img>, <math>, <object>, <svg>, <video>

MODELOS DE CONTEÚDO

Interactive content

Elementos que fazem interação com o usuário normalmente, aparecem em um formulário são ativados por um comportamento pré-determinado como clique, movimento do mouse ou entrada pelo teclado

<a>, <audio>, <button>, <details>, <embed>, <iframe>, <img>, <input>, <keygen>, <label>, <menu>, <object>, <select>, <textarea>, <video>

FORMULÁRIO E SEUS NOVOS ATRIBUTOS

Elemento input recebe novos atributos para type

- tel: sem máscara de validação e integração com agenda
- search: campo de busca semanticamente interpretado
- email: com formatação/validação e integração com agenda
- url: endereço web com formatação/validação
- color: seletor de cor renderizado pelo navegador
- number: aceita apenas valores númericos
    - step, min, max: atributos opcionais para configurar escopo
- range: variante visual do tipo number

FORMULÁRIO DATA E HORA

Novos atributos introduzidos para controle de data e hora

- datetime
- date
- month
- week
- time
- datetime-local (trata diferenças de fuso-horário) oferecem suporte a formatação/validação com o servidor
- step: define a diferença mínima entre dois horários (em s)

FORMULÁRIO USABILIDADE

Soluções de validação e usabilidade nativas ao **HTML**

- autofocus: dá foco ao campo assim que o campo for criado
- placeholder: texto padrão do campo antes do foco
- required: torna o preenchimento de um campo obrigatório
- maxlength: agora disponível para o elemento textarea
- pattern: define expressões regulares de validação (regex)
- novalidate: em form, indica a não-validação do formulário
- formnovalidate: em um botão submit, pode ser usado para submeter dados sem realizar validação (ex: rascunhos)

FORMULÁRIO CUSTOMIZAÇÃO

Ao invés de se "amarrar" ao **Javascript** e bibliotecas (como **jQuery**), o **HTML5** fornece interfaces para tornar a interação com a linguagem o mais transparente o possível

Por exemplo, para criar uma validação de dados customizada, implementamos uma rotina padronizada
- Evento oninput no input é disparado quando ocorre qualquer modificação no valor de um campo
- Método setCustomValidity, implementado dentro de um método **Javascript** e é usado para dar informar o usuário em caso de erros durante a validação

FORMULÁRIO ORTOGRAFIA

Através do uso do atributo spellcheck="true" é possível habilitar a revisão ortográfica (e também gramátical) para qualquer campo de um formulário

Vale ressaltar que, assim como a grande maioria das tags de **HTML5**, essa funcionalidade está restrita ao que estiver disponível na plataforma de destino

Também é possível desativar qualquer tipo de validação utilizando spellcheck="false"

ELEMENTO MENU

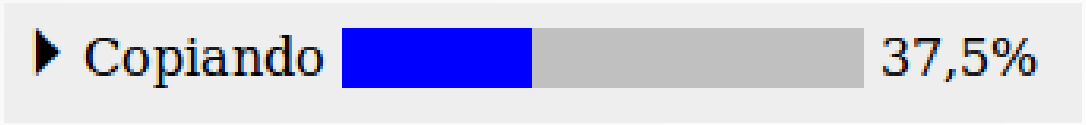
O elemento menu é usado para definir menus e barras de ferramenta de maneira simples e semântica em um navegador compátivel, exibirá os elementos menu (e seus sub-elementos) de maneira organizada e aninhada

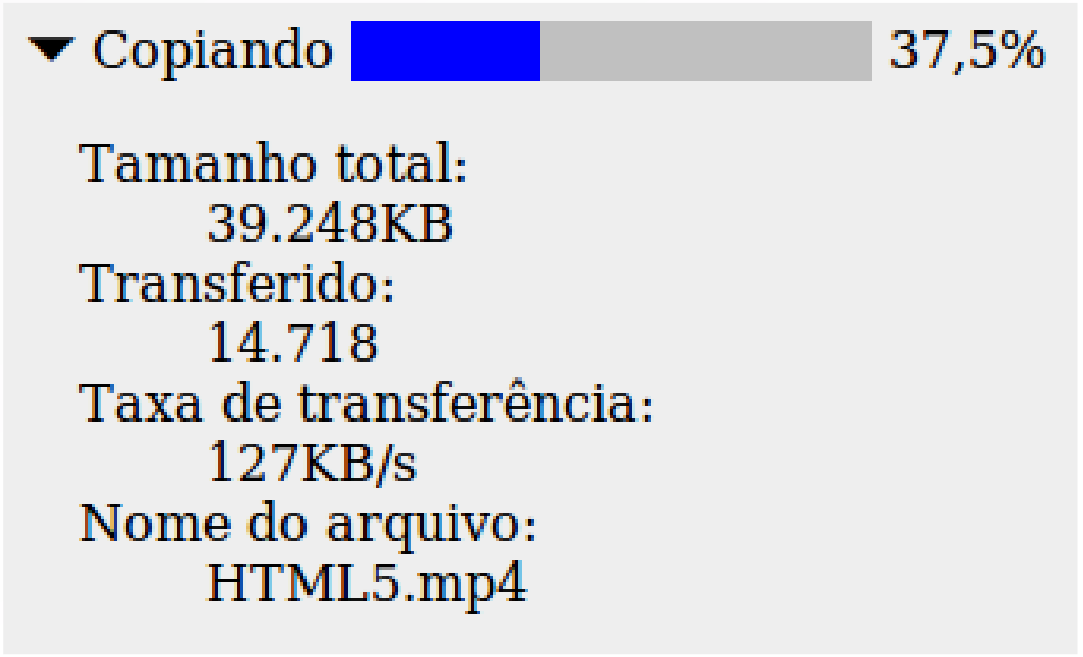
```
<menu type="toolbar">
    <li>
        <menu label="File">
            <button type="button" onclick="fnew()">New...</button>
            <button type="button" onclick="fopen()">Open...</button>
            <button type="button" onclick="fsave()">Save</button>
            <button type="button" onclick="fsaveas()">Save as...</button>
        </menu>
    </li>
    <li>
        <menu label="Edit">
            <button type="button" onclick="ecopy()">Copy</button>
            <button type="button" onclick="ecut()">Cut</button>
            <button type="button" onclick="epaste()">Paste</button>
        </menu>
    </li>
</menu>
```

- New... Open... Save Save as...
- Copy Cut Paste
- Help
- About

SUMÁRIO E DETALHAMENTO

Visualizar uma descrição sumarizada e, ao clicar, abrir o detalhamento é uma prática comum da web no entanto, isso é realizado via métodos **Javascript** no **HTML5**, foram inseridas as tags details e summary para realizar essa operação de forma semântica





CONTEÚDO EDITÁVEL

É possível fazer com que qualquer elemento do **HTML** seja editável pelo agente do usuário, para isso, basta setar o atributo contenteditable="true“.

Isso permite criar, com facilidade, uma área de edição de **HTML** dentro de uma página

TROQUES DE VISUALIZAÇÃO

Para facilitar o trabalho dos desenvolvedores foram inseridos dois antigos "truques" de **Javascript** e **CSS** foram padronizados no **HTML5**

A funcionalidade hidden agora pode ser declarada como atributo de qualquer objeto

```
<div hidden="true">Texto escondido.</div>
```

O método scrollIntoView pode ser chamado para trazer qualquer elemento da página para o foco do navegador

```
document.getElementById('id').scrollIntoView()
```

DRAG AND DROP

Essa **API** permite que elementos sejam "arrastáveis" pelo usuário – assim como ícones no sistema operacional

- No objeto arrastado, inserimos draggable="true" esse objeto dispara os eventos dragstart, drag e dragend
- No objeto alvo, não é necessário inserir nenhum atributo esse objeto escuta os eventos
  dragenter, dragleave, dragover e drop
- Fica ao cargo do programador definir o comportamento da página quando algum desses eventos for chamado

ÁUDIO, VÍDEO AND CODECS

Novos elementos substituem o uso de elementos iframe ou embed para renderizar players de áudio

- Elementos audio e video podem ser customizados
- Controls: define a exibição de uma barra de controle
- Autoplay: define se o conteúdo terá reprodução automática
- Source: tags utilizadas para definir fontes alternativas
- Codecs: informa ao browser qual o formato da fonte alternativa

```
<video controls="true" autoplay="true" width="400" height="300">
        <source src='video.ogv' type='video/ogg; codecs="theora, vorbis"'>
        <source src='video.mp4' type='video/mp4; codecs="mp4v.20.240, mp4a.40.2"'>
        <p>Faça o <a href="video.mp4">download do vídeo</a></p>
</video>
```

MATHML

MathML é uma linguagem de marcação, baseada em **XML**, usada para representação de fórmulas matemáticas no **HTML5**, o elemento **math** denota o uso de **MathML**

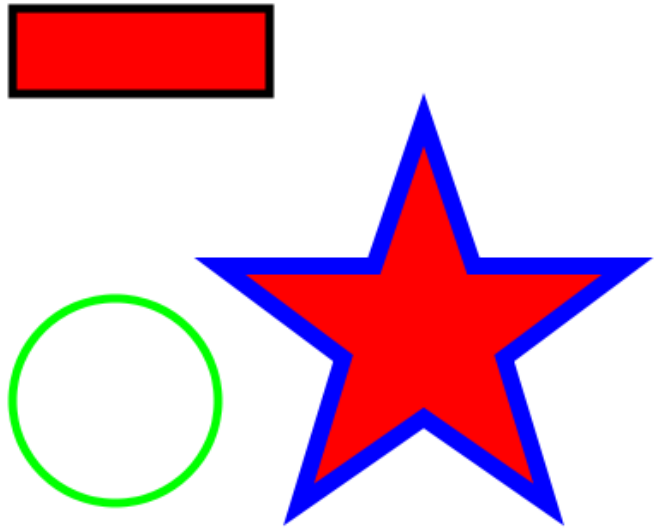
```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mi>x</mi>
    <mo>=</mo>
    <mfrac>
      <mrow>
        <mo form="prefix">&minus;</mo><mi>b</mi>
        <mo>&PlusMinus;</mo>
        <msqrt>
          <msup>
            <mi>b</mi><mn>2</mn>
          </msup>
          <mo>&minus;</mo>
          <mn>4</mn><mo>&InvisibleTimes;</mo><mi>a</mi><mo>&InvisibleTimes;</mo><mi>c</mi>
        </msqrt>
      </mrow>
      <mrow>
        <mn>2</mn><mo>&InvisibleTimes;</mo><mi>a</mi>
      </mrow>
    </mfrac>
  </mrow>
</math>
```

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

SVG

SVG é uma linguagem de marcação, baseada em **XML**, usada para marcação de gráficos vetoriais no **HTML5**, o elemento **svg** denota o uso de **SVG** o conteúdo vetorial é escalável e acessível a leitores de tela

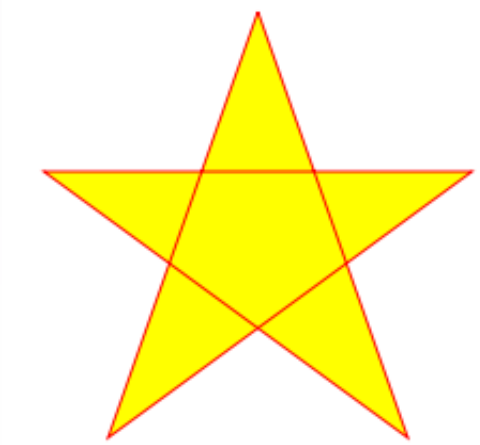
```
<svg width="400" height="400">
        <!-- Um retângulo: -->
        <rect x="10" y="10" width="150" height="50" stroke="#000000" stroke-width="5" fill="#FF0000" />
        <!-- Um polígono: -->
        <polygon fill="red" stroke="blue" stroke-width="10"
                points="250,75 279,161 369,161 297,215
                        323,301 250,250 177,301 203,215
                        131,161 221,161" />
        <!-- Um círculo -->
        <circle cx="70" cy="240" r="60" stroke="#00FF00" stroke-width="5" fill="#FFFFFF" />
</svg>
```



CANVAS

Canvas API permite a criação de desenhos na tela do navegador usando **Javascript** para renderização do desenho, usamos um elemento **canvas** performance superior ao **SVG** e sintaxe simplificada

```
function desenhar(){
        context=document.getElementById('x').getContext('2d')
        //Iniciamos um novo desenho
        context.beginPath()
        //Movemos a caneta para o inicio do desenho
        context.moveTo(150,50)
        //Desenhamos as linhas
        context.lineTo(220,250)
        context.lineTo(50,125)
        context.lineTo(250,125)
        context.lineTo(80,250)
        context.lineTo(150,50)
        //Vamos pintar o interior de amarelo
        context.fillStyle='#ff0'
        context.fill()
        //Vamos pintar as linhas de vermelho.
        context.strokeStyle='#f00'
        context.stroke()
}
```



HISTÓRICO DE SESSÃO

Anteriormente, os browsers não tinham controle sobre o histórico de ações de um usuário em uma página – limitando-se a navegação usando os métodos go, back e forward do objeto history do navegador

O **HTML5** turbinou o objeto history com novos métodos para controlar a pilha de entradas do histórico e também associar dados à essas entradas com maior liberdade

- pushState(data, title, url): acrescenta entrada no histórico
- replaceState(data, title, url): modifica entrada atual
- window.onpopstate : evento ativado ao navegar no histórico

STORAGE

Ao invés de trabalhar com cookies complexos e com funcionalidade limitada, o **HTML5** traz uma nova maneira de armazenar (e recuperar) dados no clientes – a **API Storage**

Existem dois objetos de que podemos implementar

- localStorage: armazena dados sem data de expiração
- sessionStorage: armazena dados apenas durante a sessão

Interface simplificada de acesso aos dados

getItem(key) setItem(key, value), removeItem(key) clear()

GEOLOCATION

A **Geolocation API** é capaz de permitir que o navegador detecte a posição geográfica de um cliente possível pelo **IP**, triangularização de antenas **GPRS** ou **GPS**

A documentação especifica que o navegador deve perguntar ao usuário se ele deseja compartilhar sua localização

Como, por uma série de motivos, esses dados podem não estar disponíveis, é necessário cautela na implementação

```
navigator.geolocation.getCurrentPosition(showpos)
function showpos(position){
  alert('Your position: '+position.coords.latitude+','+ position.coords.longitude )
}
```

AS TAGS DO HTML5 – breve explicação

<canvas>
<details>
<summary>
Titulo de uma seção details

<header>
Título de uma section. Pode conter headings, hgroup, Content table, logos, search, form...

OBS: O hgroup foi descontinuado em abril/13.

AS TAGS DO HTML5 – breve explicação

<section>
Agrupa itens que possuam relação entre si, e cujo agrupamento
Não seja meramente para fins de posicionamento

<footer>
Rodapé, geralmente é o último elemento do documento

<article>
Conteúdo agrupado, importante passível de utilização em feeds

<figure>
Uma figura, possivelmente possuirá legenda

<figcaption>
Legenda de uma figura

AS TAGS DO HTML5 – breve explicação

```
<datalist>

<input list="browsers">

<datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
</datalist>
```

AS TAGS DO HTML5 – breve explicação

<mark>
Realce partes de um texto

<meter>
Medidor em formato de barra. Atributos: min, max, value

<nav>
Tipo de section que agrupa links de navegação

<aside>
Citação, sidebars, anúncios publicitários e grupos de navegação

AS TAGS DO HTML5 – breve explicação

<output>
Representa o resultado de cálculo

```
<form>
    <input type="range" id="a" value="50">+
    <input type="number" id="b" value="50">=
    <output name="x" for="a b"></output>
</form>
```

AS TAGS DO HTML5 – breve explicação

<progress>
Barra de progresseo, usada para representar o progresso de uma determinada ação. Pode ser um upload de uma imagem

<time>
Representa uma data, hora ou ambos (atributo datetime)

TAGS DEPRECADAS

<acronym>
<applet>
<basefont>
<big>
<canter>
<font>
<frame>
<frameset>
<noframes>
<strike>
<tt>
...

CSS3 - SELETORES

Element1~element2[sequênciais]

[attribute^=value]
Inicia exatamente com a string value

[attribute$=value]
Termina exatamente com a string value

[attribute*=value]
Contém a string value

CSS3 - SELETORES

:first-of-type
:last-of-type
:only-of-type
:only-child
:nth-child(n)
:nth-last-child(n)
:nth-of-type(n)
:nth-last-of-type
:last-child
:root
:empty (sem filhos, nem ao menos text-nodes)
:enabled
:disabled
:checked
:not(selector)
::selection

CSS3 - PROPRIEDADES

background-clip

padding-box, content-box, border-box, text

background-position

Left top, left center, left bottom, 50% 50%, 10px 200px

background-origin

padding-box, content-box, border-box

CSS3 - PROPRIEDADES

background-size

width, height

200px
200px auto
50% 50%
container e cover

CSS3 - PROPRIEDADES

border-radius

```
border-radius: 10px;
border-top-left-radius:25px;
```

CSS3 - PROPRIEDADES

box-decoration-break
slice
Clone

box-shadow
hr vr blur color

overflow-x
visible, hidden, scroll, auto

overflow-y
visible, hidden, scroll, auto

CSS3 - PROPRIEDADES

Rotate (CSS transformations)
transform: rotate(60deg)

opacity
opcity: 0.5;

text-shadow
hr, vr, blur, color

transition (background-color)
800ms ease;

Linear, ease, ease-in, ease-out

PRÉ-PROCESSADORES CSS

SASS

LESS

STYLUS

REFERÊNCIAS

http://www.w3.org

http://www.infowester.com/introhtml5.php

https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content_categories

http://diveintohtml5.info/video.html

http://www.modernizr.com

Se você chegou até aqui, é porque realmente quer aprender a programar do jeito certo.

Poucos começam com essa mentalidade. E é ótimo que você esteja dentro desse grupo seleto de pessoas que desejam fazer as coisas da melhor maneira possível!

PARABÉNS!

Gostaria de agradecer você por ter lido este E-Book de **HTML5 e CSS3 do Jeito Certo**.

Espero que você tenha gostado e achado útil. Mais que isso, espero que você utilize essas dicas no seu dia-a-dia de programação.

E o Que Fazer Agora...?

O que eu mostrei pra você aqui é só o começo. Há muitas técnicas de programação lá fora! E de nada adianta conhecer as boas práticas sem conhecer bem a linguagem com a qual estamos trabalhando.

Por isso você precisa aprender a programar em **HTML5** e **CSS3** pra colocar em prática as técnicas e dicas que aprendeu neste **E-Book**.

Meu nome é **Luiz Augusto Vieira**, sou programador **Front-end** desde 2009, sou apaixonado pela profissão e por isso sempre que posso ajudo quem está afim de ingressar na área.

Até setembro/2015 eu trabalhava em uma Agência de publicidade, das 09h às ?, ou seja, só tinha hora para entrar. Pois quem trabalha em Agência sabe do que estou falando.

Mas para quem está buscando experiências precisa passar por isso, pois, faz parte da nossa vida no dia a dia.

Mas, uma hora você vai querer mudar, foi o que aconteceu comigo, eu procurei minha liberdade financeira e mais qualidade de vida. Decidir que era a hora de sair da Agência e trabalhar em casa por conta própria, fazer meu horário de trabalho e ter tempo para passear e ajudar alguém da área ao meu alcance.

Resumindo, me desliguei da Agência em setembro/2015 e estou trabalhando em casa com todo o conforto que eu procurava e que eu mereço.

BLOG:
https://www.htmlecsspro.com

Youtube:
https://www.youtube.com/htmlecsspro

Facebook:
https://www.facebook.com/htmlecsspro

Contato:
contato@htmlecsspro.com

AO SEU SUCESSO.
OBRIGADO!!!

- LUIZ AUGUSTO VIEIRA