

# Programação para a Web

## Tópico 3 - Linguagens de Marcação

Prof. Ricardo Masao Kondo

[ricardo.kondo@ufms.br](mailto:ricardo.kondo@ufms.br)

Slides cedidos Prof. Dr. Hudson Silva Borges

# Material Disponibilizado



# Linguagens de Marcação

- “Uma linguagem de marcação combina texto e informação extra sobre o texto”
- Exemplos de linguagens de marcação:
  - **SGML**
    - *Standard Generalized Markup Language*
    - ISO 8879:1986
  - **HTML**
    - *Hypertext Markup Language*
    - RFC 1866 (HTML 2.0)

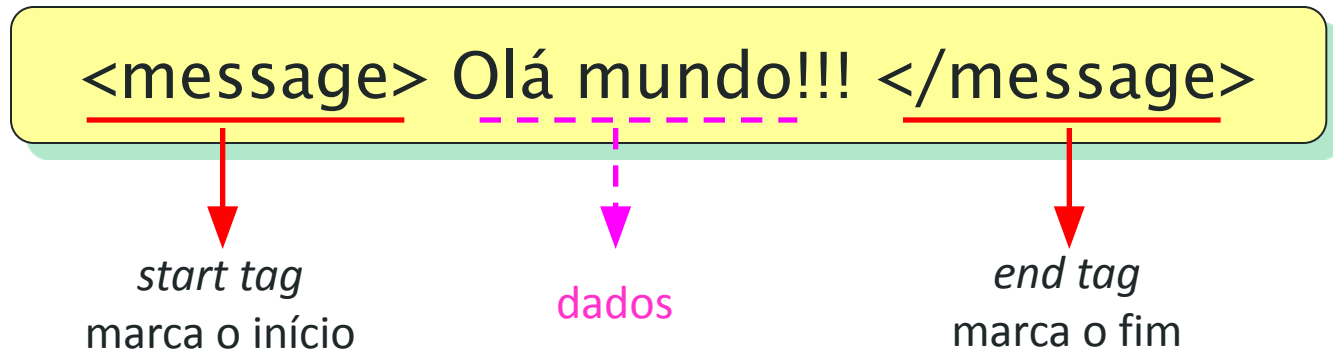
# Linguagens de Marcação

## Tags (etiquetas)

- *descriptive markups*
- Marcação que delimita um elemento (SGML)

## Elemento

- *Um componente da estrutura de um documento*



# Linguagens de Marcação

- Exemplo:

```
<html>
  <head>
    <title>Exemplo 1</title>
  </head>
  <body>
    O meu <em>número</em> é <strong>54</strong> e eu vou
    ganhar o Prêmio.
  </body>
</html>
```

# Linguagens de Marcação

- **HTML**

- Define uma classe simples de documentos
  - Cabeçalhos, parágrafos, listas, tabelas e imagens
- Suporte mínimo a hipertexto
  - Ligação unidirecional especificada integralmente dentro do documento

- Outras limitações

- A linguagem não é extensível
- Um documento não pode ser reutilizado
- Pouca, ou quase nenhuma semântica pode ser extraída de um documento

# Linguagens de Marcação

- **SGML**

- Linguagem bastante extensa e complexa
  - Dificulta a criação de aplicações (*tools*) para o ambiente Web
- HTML é uma aplicação da SGML

- Vantagem

- Flexibilidade
  - Definição dos elementos necessários em cada aplicação
  - Definição de novas aplicações quando necessárias

- Desvantagem

- Custo do processamento no ambiente Web

# Linguagens de Marcação

- XML

- **Extensible Markup Language**
- Um *application profile* da SGML
- Omite todas as partes opcionais e diversas partes mais complexas e menos usadas da SGML

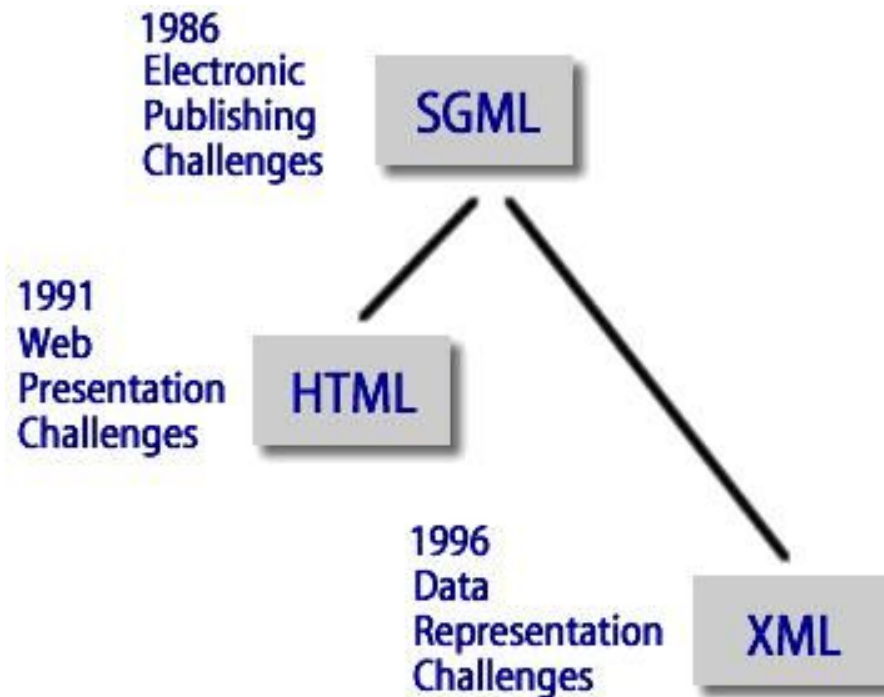
- Vantagem

- Facilidade
  - Definir tipos de documentos
  - Escrever programas para manipular os documentos



# Linguagens de Marcação

- Qual a relação entre elas???



# HTML

**<meta>**

**HyperText Markup  
Language**

**</meta>**



# HTML: O que é?

- HTML = **H**yper**T**ext **M**arkup **L**anguage
- “Linguagem de marcação de hipertextos”
  - Permite a ligação entre uma porção de texto de um documento e outro documento
  - Têm a vantagem de dar contexto e de facilitar uma sequência natural de consulta (Estrutura não linear)
  - Criar formulários que permitem a introdução de dados

## HTML: O que não é?

- **Não deve ser usado como meio de definir o estilo de um documento** (fontes, cores, contornos, ...)
- Não deve ser usado para definir o posicionamento de elementos num documento
- **Separar conteúdo da apresentação**

# URL e HTTP

- URL = **U**niform **R**esource **L**ocator
  - A localização de um qualquer recurso é definida num URL
    - *serviço://computador/arquivo*
    - *serviço* indica o protocolo do servidor (http, ftp, ...)
    - *computador* é a máquina onde corre o servidor
    - *arquivo* é o nome completo do arquivo pretendido
- HTTP = **H**yper**T**ext **T**ransfer **P**rotocol
  - O protocolo que é usado para transferir documentos HTML que estão disponíveis numa dada URL

# Surgimento do Hipertexto

- Vannevar Bush (1945)
  - “*As We May Think*” - *Memex*
- Ted Nelson (1965)
  - Termo Hipertexto
  - Projeto Xanadu -> mundo virtual de docs

# História do HTML

- (1991) HTML foi definido
  - Tim Berners-Lee
- (1993) Surgimento do *browser* MOSAIC
- (1993 - 1997) O HTML sofreu uma série de extensões
  - HTML 2.0 da IETF, HTML+, HTML 3.0

# História do HTML

- (1996) A W3C (*World Wide Web Consortium*) lançou o HTML 3.2.
- (1997) HTML 4.0 pela W3C
- (1999) HTML 4.01
  - Correções de erros do HTML 4.0.
- (2000) XHTML: e**X**tensible **H**yper**T**ext **M**arkup **L**anguage
  - Revisada em 2002



# História do HTML

- (2004) HTML 5 – Nascimento...
- depois de um workshop do W3C, Apple, Mozilla e Opera estavam preocupadas a respeito da direção do XHTML no W3C, perda de interesse no HTML e aparente indiferença para com as necessidades dos autores do mundo real
  - Essas organizações ajustaram-se com a missão de enfrentar essas preocupações e nasceu o WHATWG (The **W**eb **H**ypertext **A**pplication **T**echnology **W**orking **G**roup)



# História do HTML

- (2008) HTML 5 (first draft)
  - W3C Working Draft 22 January 2008
- (2010) XHTML 2.0
  - Diversos working drafts de 2002 a 2006
  - <https://www.w3.org/TR/xhtml2/>
- (2012) HTML 5 (first CR)
  - W3C *Candidate Recommendation* 17 December 2012
  - <http://www.w3.org/TR/2012/CR-html5-20121217/>

# História do HTML

- (2014) HTML 5
  - **W3C Recommendation 28/Out/2014**
    - <http://www.w3.org/TR/html5/>
  - (Nov/2016) HTML 5.1
  - (Dez/2017) HTML 5.2
    - <https://www.w3.org/TR/2017/REC-html52-20171214/>
  - (Jul/2018) HTML 5.3 (Working Draft)
    - <https://www.w3.org/TR/html53/>

# História do HTML

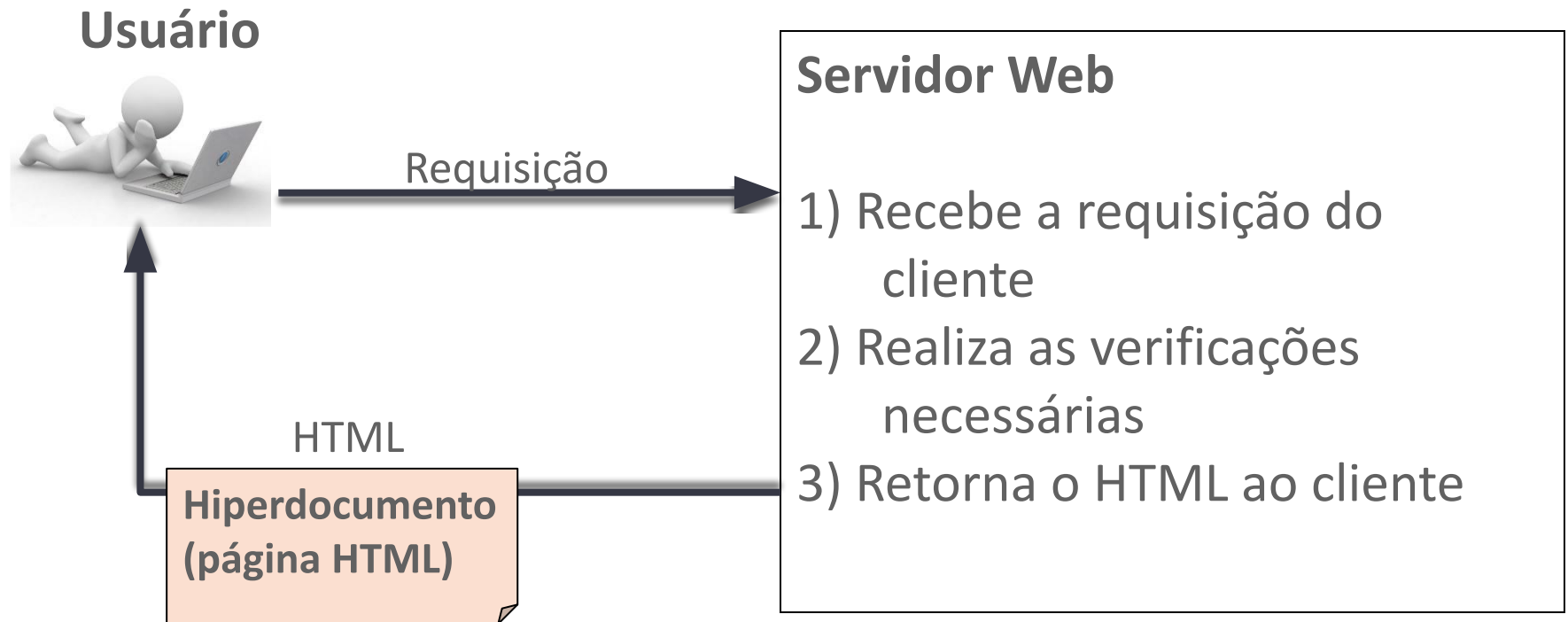
- Desde 2017 o WHATWG é formado por Apple, Google, Mozilla e Microsoft
  - <https://whatwg.org/>
  - é uma crescente comunidade de fornecedores de browsers, desenvolvedores web, e outras pessoas interessadas



# História do HTML

- HTML é fruto do "casamento" dos padrões **HyTime** e **SGML**
  - **HyTime** é um padrão para a representação estruturada de hipermídia e conteúdo baseado em tempo
    - Um documento é visto como um conjunto de eventos concorrentes dependentes de tempo (como áudio, vídeo, etc.), conectados por hiper-ligações
  - **SGML** é um padrão de formatação de textos
    - Não foi desenvolvido para hipertexto, mas tornou-se conveniente para transformar documentos em hiper-objetos e para descrever as ligações

# Requisições

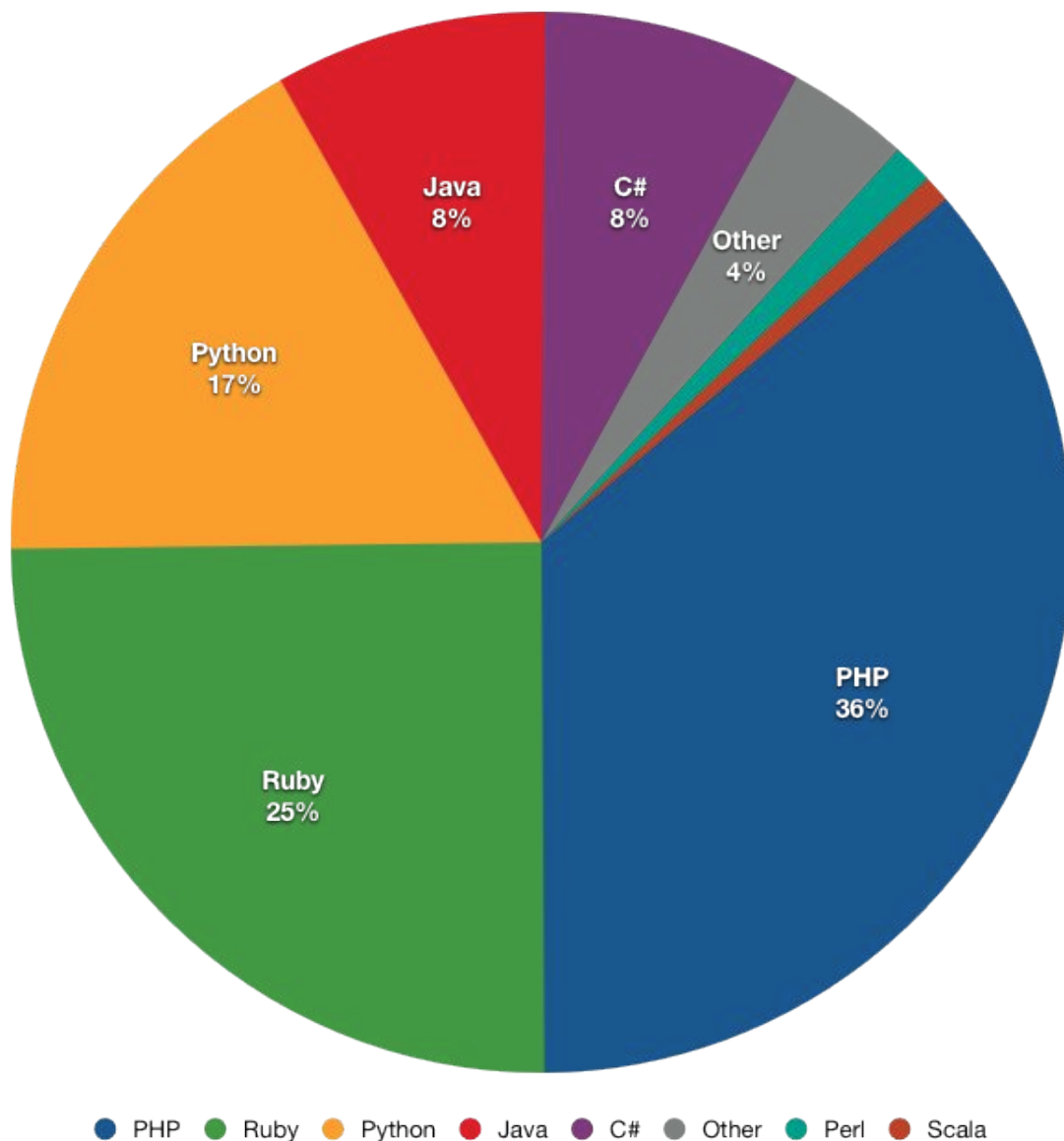


# ***Server-Side***

- Web Server
- Arquivos HTML, CSS, Imagens, Áudio, etc...
- Principais Linguagens
  - ASP (*Active Server Pages*)
  - PHP (*Hypertext PreProcessor*)
  - JSP (*JavaServer Pages*)
- Bases de Dados
- Responde às requisições dos clientes

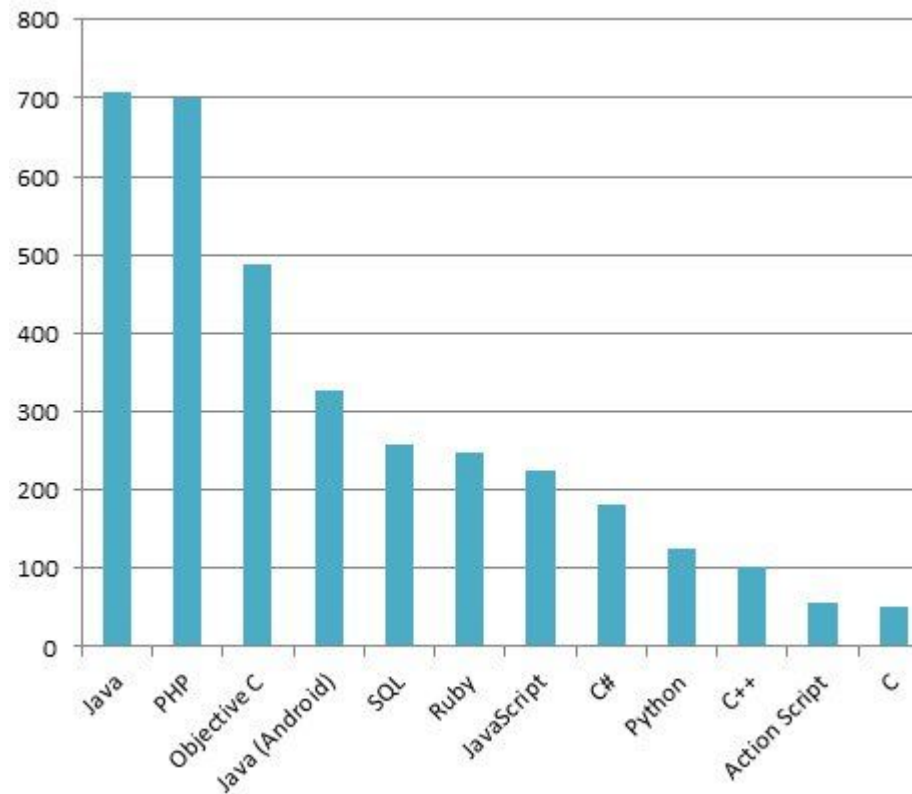
# Lado do Servidor

- Quantidade de aplicações desenvolvidas e suas respectivas linguagens em 30/05/2010.
- Dados obtidos dos clientes de [directedge.com](http://directedge.com)





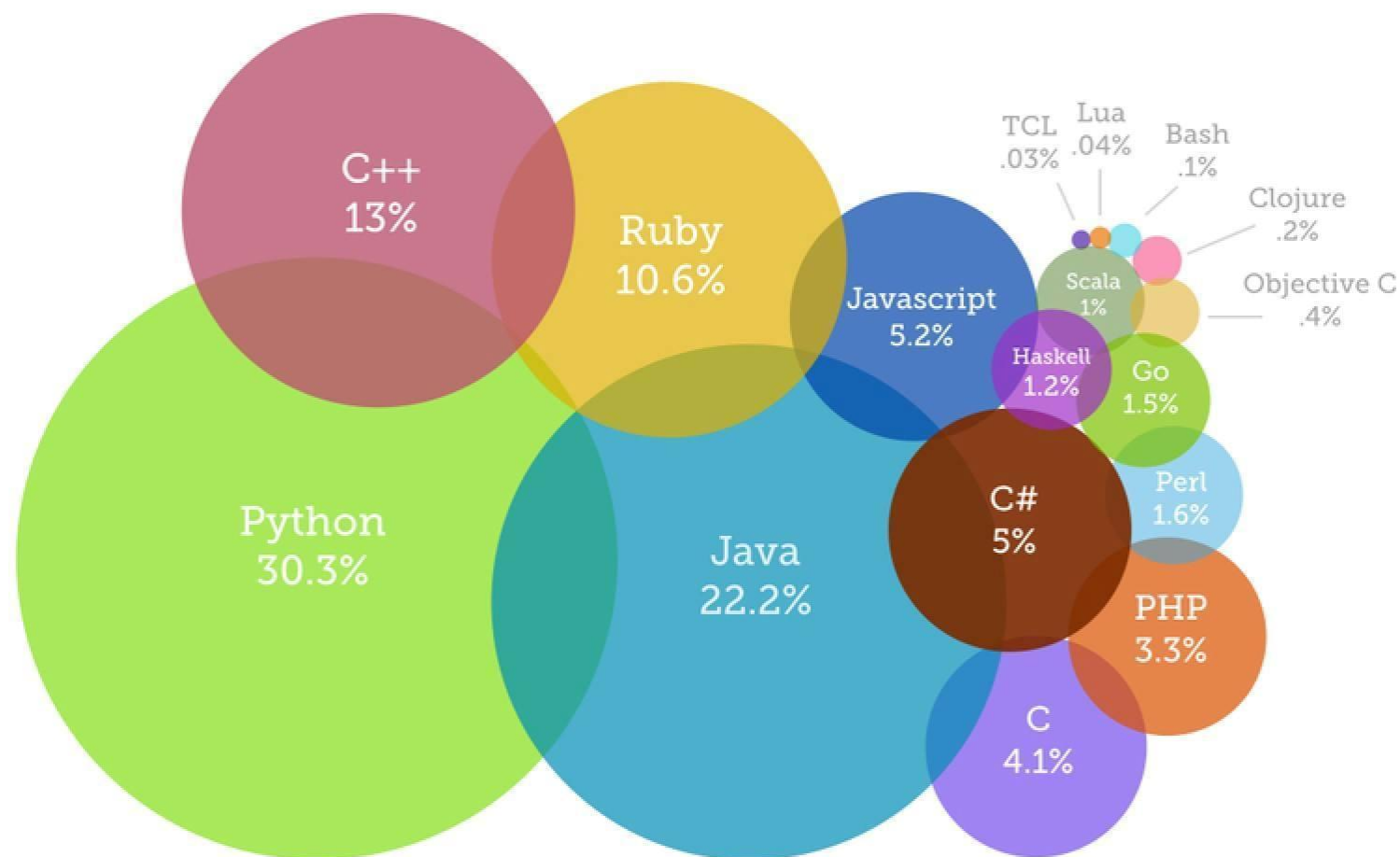
# Comparação de Linguagens



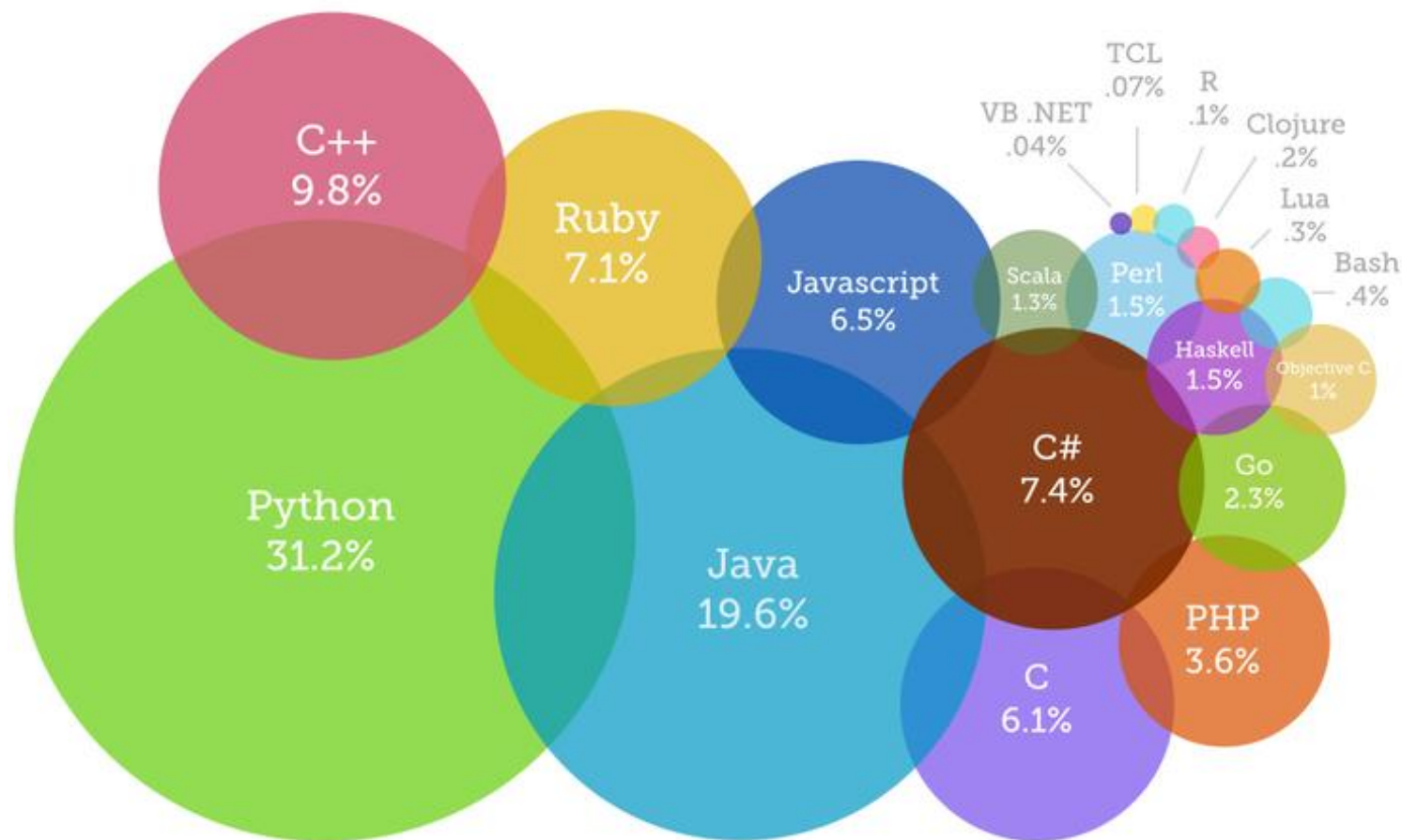
Número de vagas em um website de empregos americano (2013)

Fonte: <http://themarble.co.uk/top-10-programming-languages>

# Dados de 2014

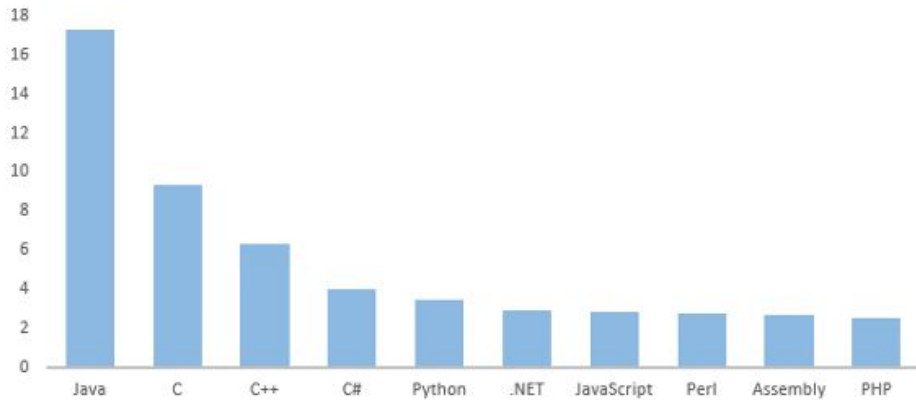


# Dados de 2015

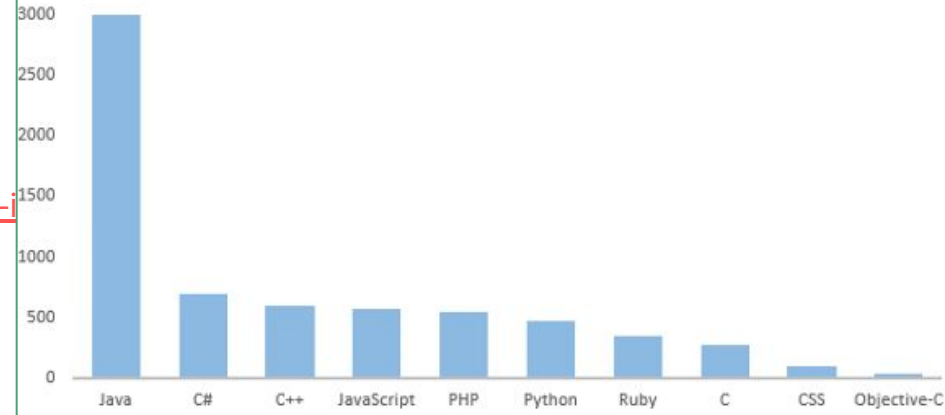


# Dados de 2017

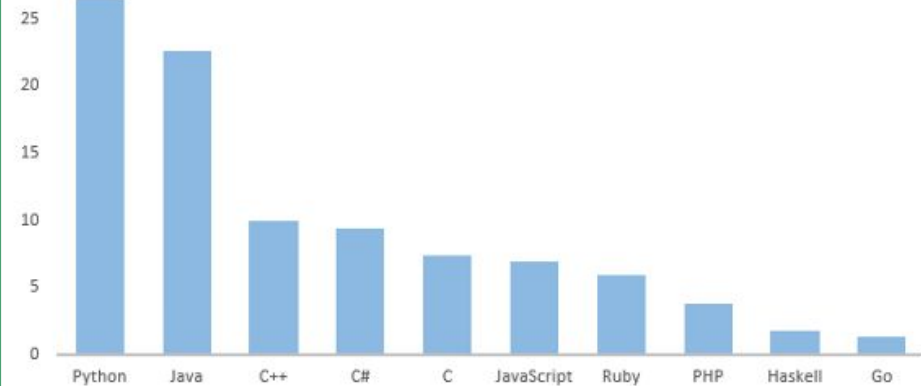
Tiobe Index, %



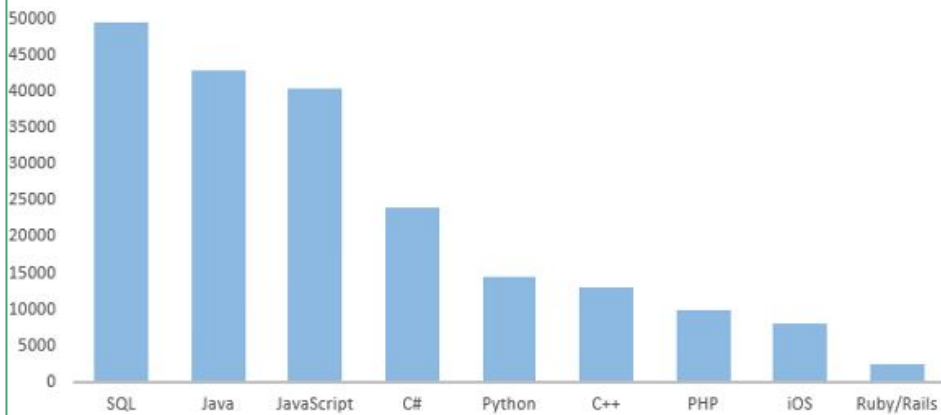
New Relic Rating



CodeEval Rating, %

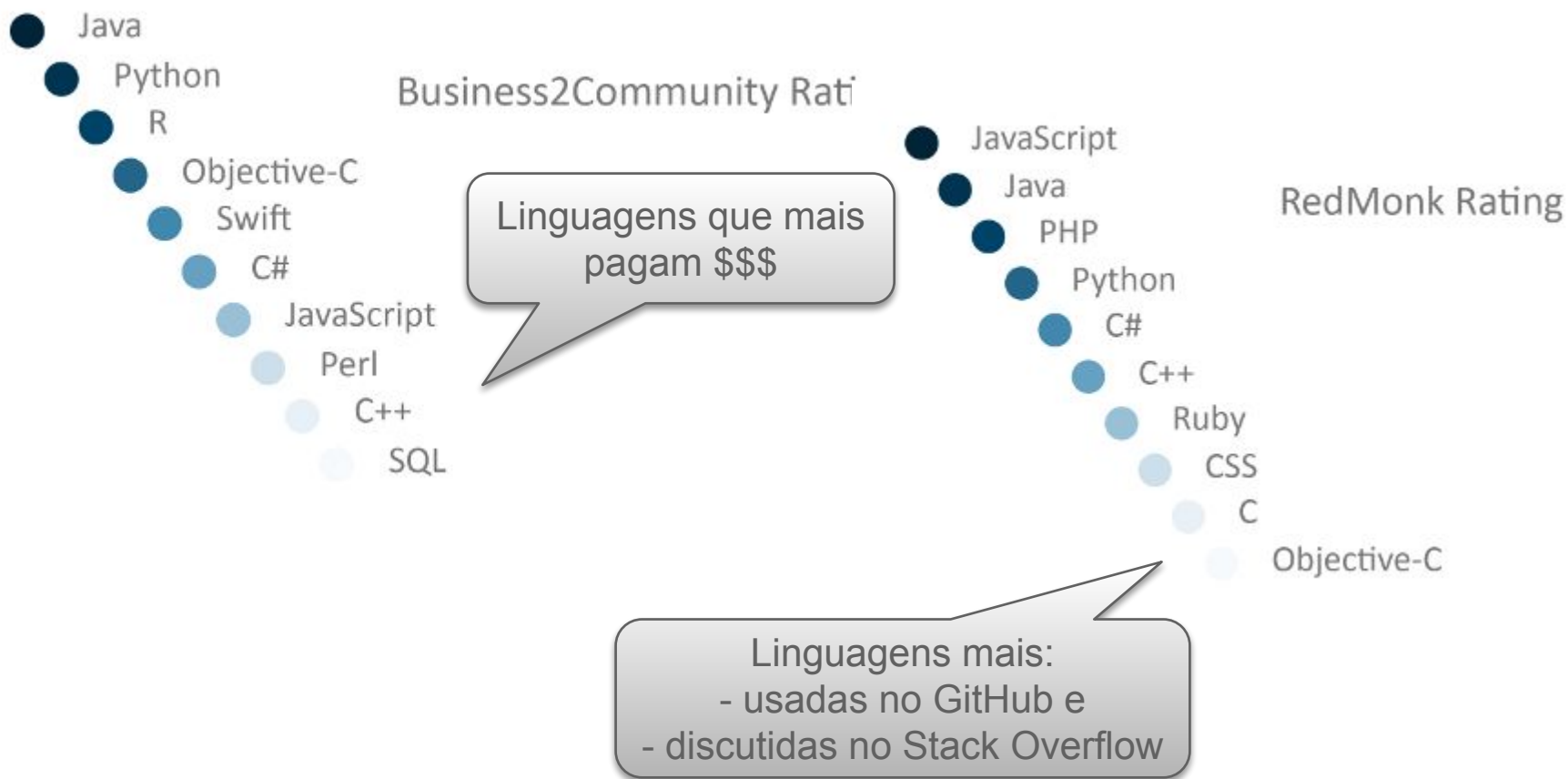


Coding Dojo Rating



# Dados de 2017

- <https://www.qualium-systems.com/blog/useful-it-articles-and-advice/top-10-programming-languages-that-would-be-most-in-demand-in-2017/>

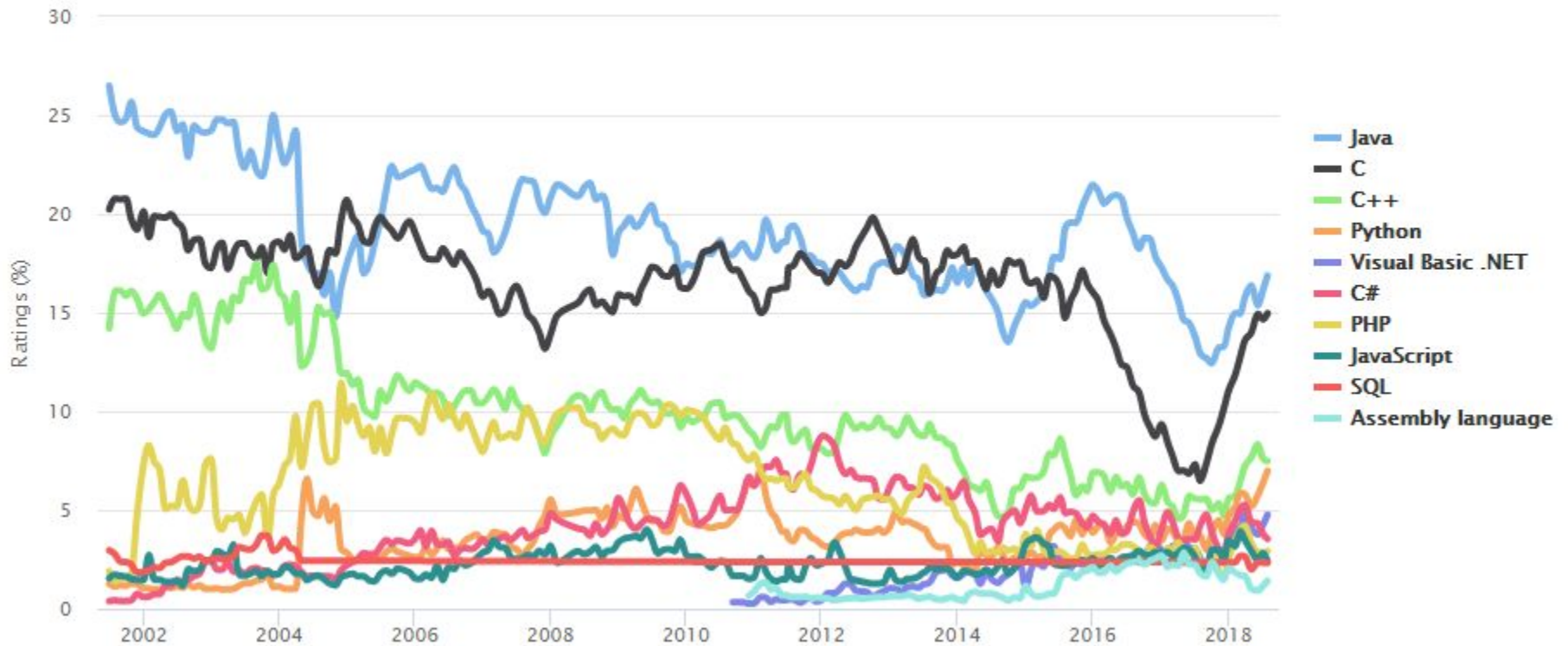


# Dados de 2018

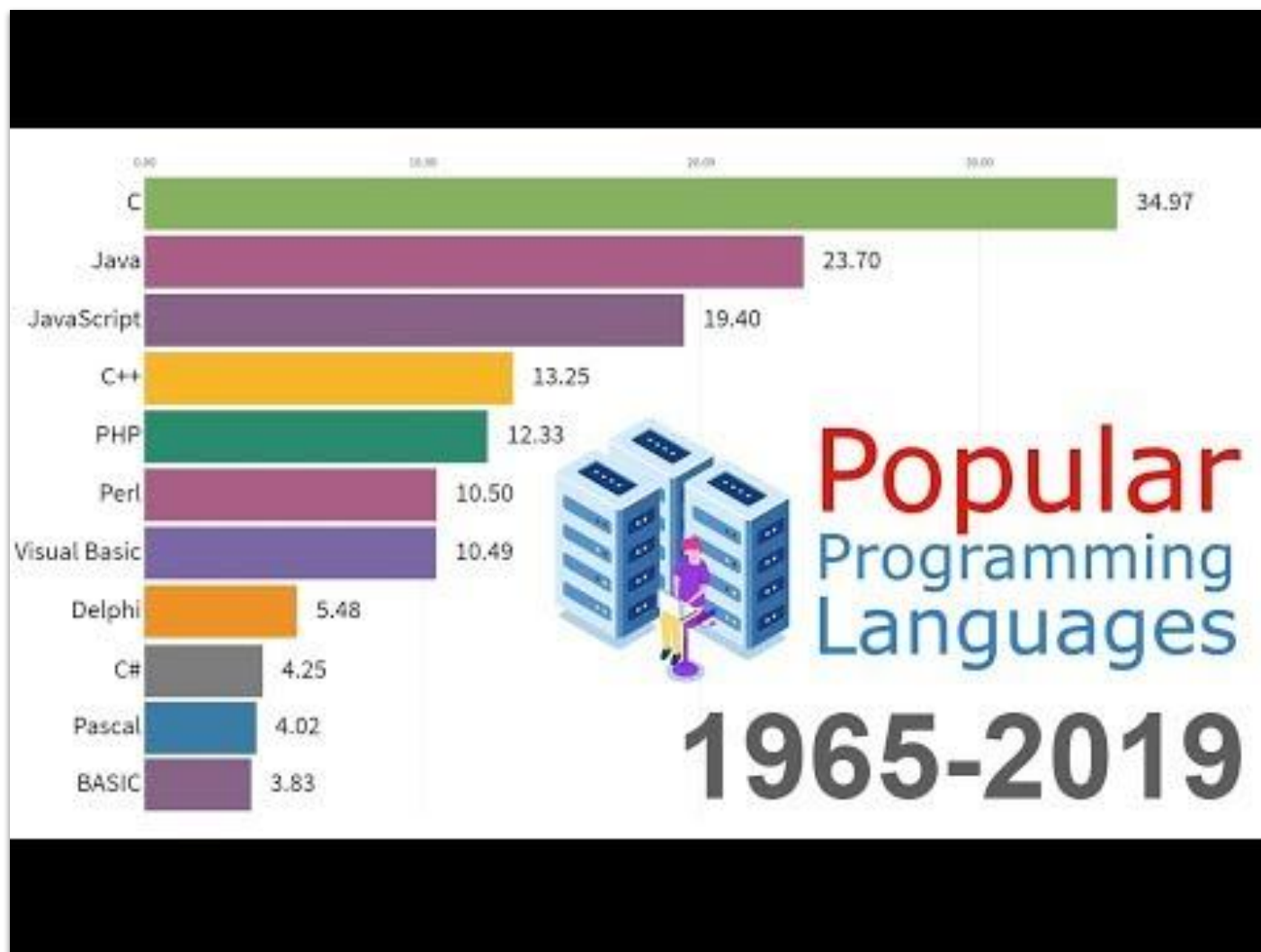
- <https://www.tiobe.com/tiobe-index/>



Source: [www.tiobe.com](http://www.tiobe.com)



# Linguagens mais populares



# *Client-Side*

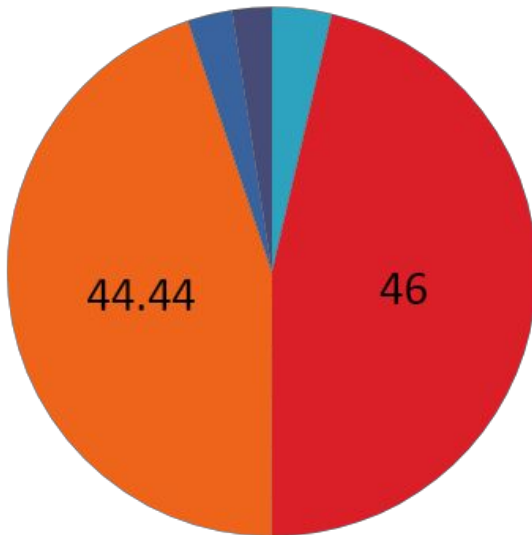
- Navegadores Web
  - Mozilla Firefox, Chrome, Edge e outros
- Linguagens
  - JavaScript
  - HTML
  - Ajax
- Faz requisições ao servidor



# Client-Side

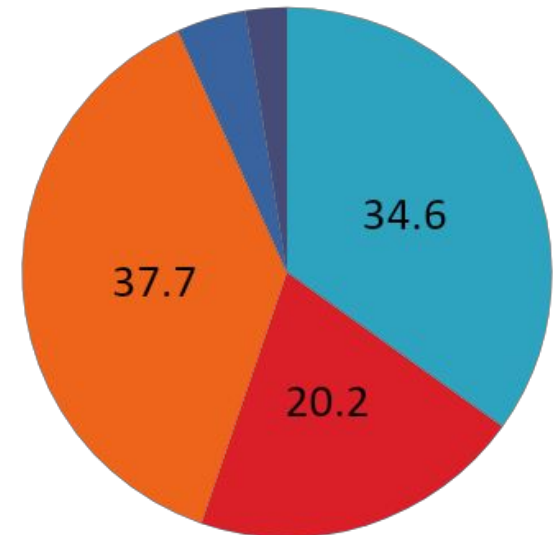
## Browser Statistics (%) - 2008

■ Chrome ■ IE ■ Firefox ■ Safari ■ Opera  
2.7 2.4 3.6



## Browser Statistics (%) - 2011

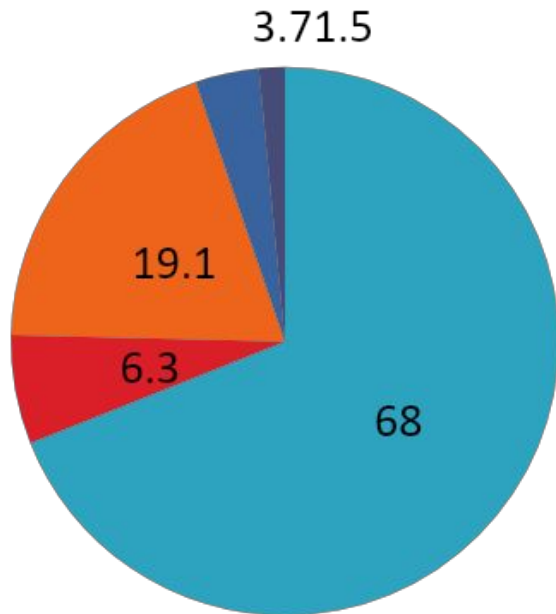
■ Chrome ■ IE ■ Firefox ■ Safari ■ Opera  
4.2 2.2 5



# Client-Side

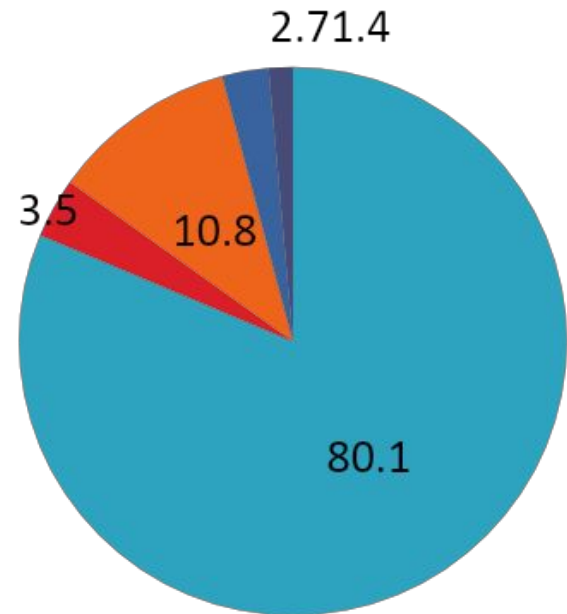
## Browser Statistics(%) - 2015

■ Chrome ■ IE ■ Firefox ■ Safari ■ Opera



## Browser Statistics(%) - 2018

■ Chrome ■ IE/Edge ■ Firefox ■ Safari ■ Opera



# Client-Side

<b>2023</b>	<b><u>Chrome</u></b>	<b><u>Edge</u></b>	<b><u>Firefox</u></b>	<b><u>Safari</u></b>	<b><u>Opera</u></b>
February	79.7 %	8.6 %	4.8 %	3.9 %	2.2 %
January	79.7 %	8.2 %	5.0 %	3.9 %	2.3 %

# HTML 4.01

Um pouco de cada...

# HTML 5

---

# Estrutura HTML

- Linguagem HTML
  - Conjunto de *Tags* e atributos
- Estruturada
- Interpretada pelo *Browser*
  - Que mostra a página HTML para o usuário
- Extensões: `.html` ou `.htm`

# Estrutura HTML – *Tags*

- Todo documento HTML é formado por tags (etiquetas)
  - Delimitada pelos caracteres < e >  
    <etiqueta>...</etiqueta>
- Uma tag é formada por comandos, atributos e valores

# Estrutura HTML – *Tags*

- Exemplo:

The **<abbr title=" HyperText Markup Language 5">  
HTML 5</abbr>** was recommended in 2014.

- No qual:

- **abbr** = tag que representa um acrônimo/sigla/abreviatura
- **title** = atributo que especifica o título (significado) da abreviatura
- **“HyperText Markup Language 5”** = valor do atributo title, que é o significado da abreviatura citada na tag

# Estrutura HTML – *Tags*

- Comentários
  - Delimitado pelos caracteres `<!--` e `-->`
- Abertura e Fechamento (pares)
  - Exemplos
    - `<html>` e `</html>`
    - `<table>` e `</table>`
    - `<div>` e `</div>`
    - `<!-- Isto é um comentário -->`
  - Exceções
    - `<br>`, `<img>`



# Estrutura HTML – *Atributos*

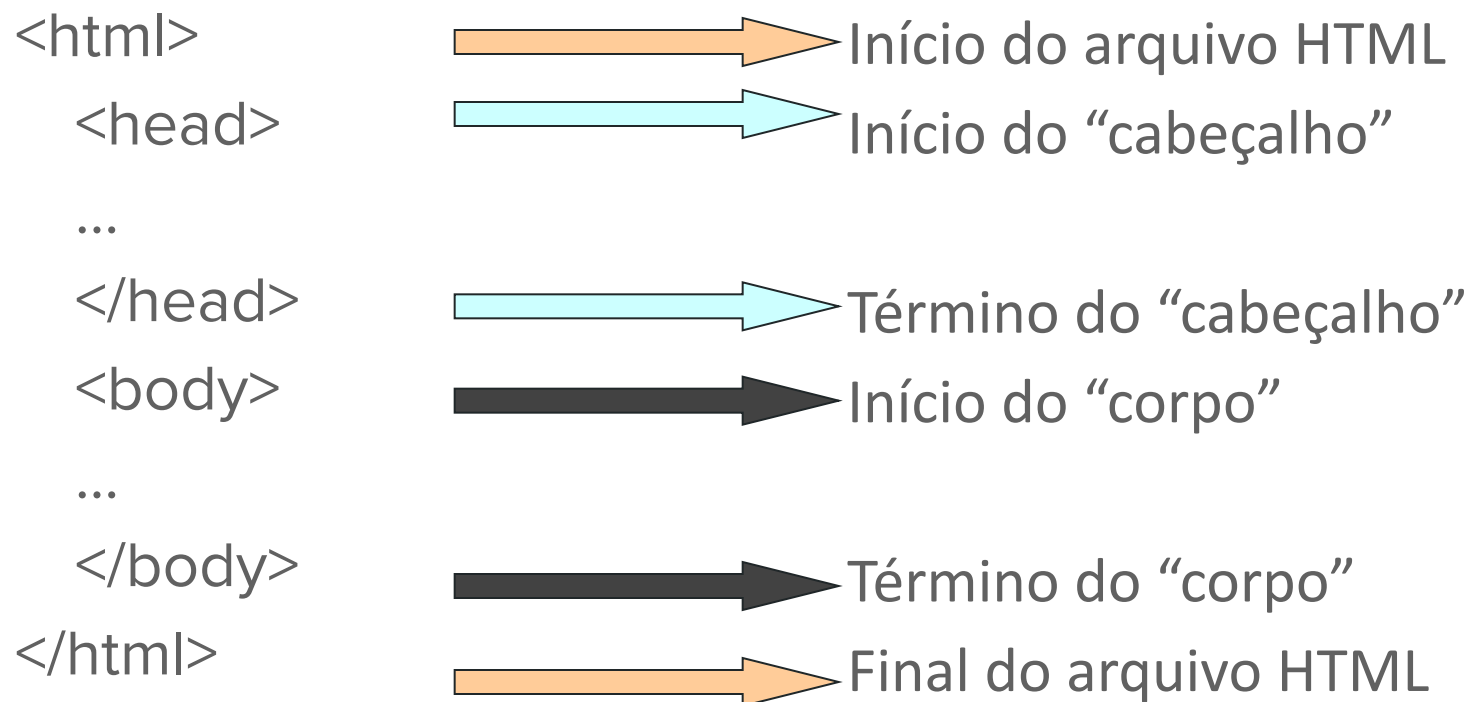
- Modificam as *Tags*
- Devem estar dentro das *Tags* (de abertura)
- Separadas sempre por um espaço
- Possuem valores
  - Devem estar entre aspas duplas " "
  - Formato nome="valor"

# Estrutura HTML – *Atributos*

- Exemplos
  - `<option value="estudante">Estudante</option>`
  - `<input type="text" size="40">`

# Estrutura HTML – *Documento*

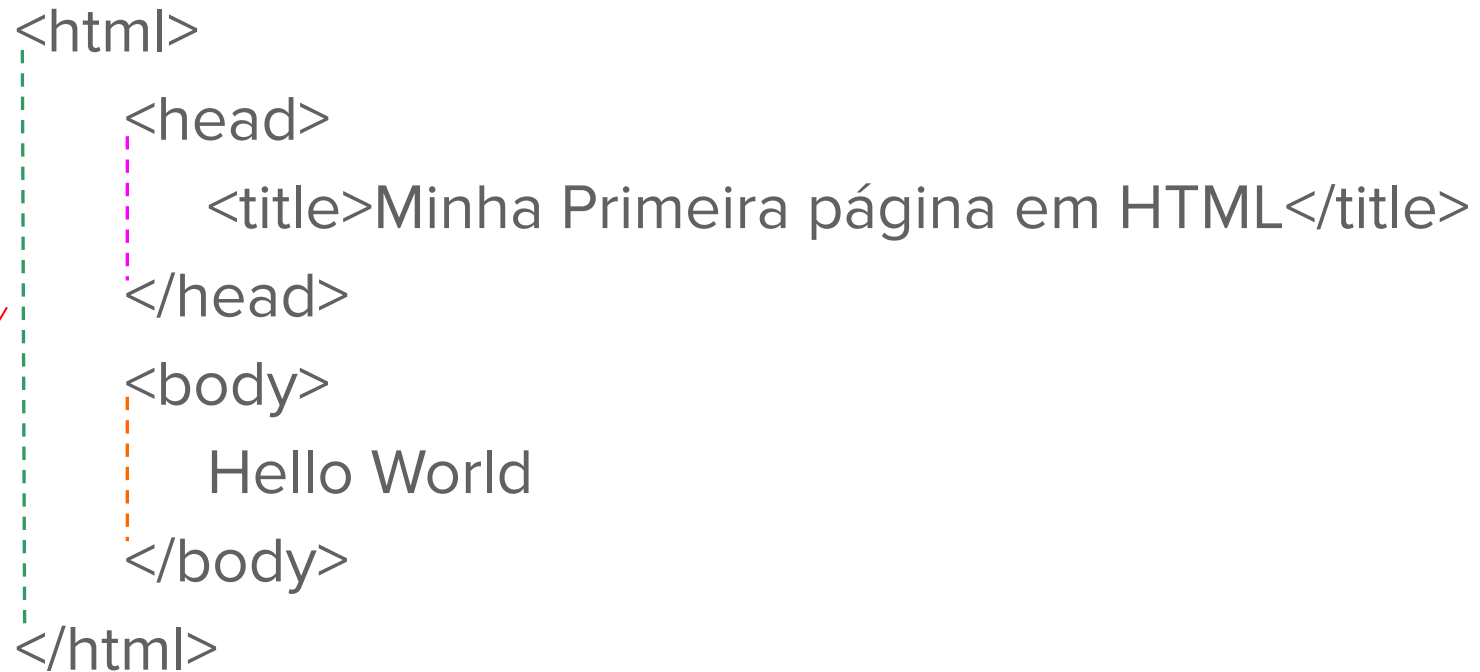
- Estrutura básica de arquivos HTML:



# Exemplo

- *Hello World* em HTML

```
<html>  
  <head>  
    <title>Minha Primeira página em HTML</title>  
  </head>  
  <body>  
    Hello World  
  </body>  
</html>
```

The diagram illustrates the hierarchical structure of an HTML document. It uses colored dashed lines to show the nesting of tags: a green dashed line for the root <html> tag, a magenta dashed line for the <head> tag, and an orange dashed line for the <body> tag. A red arrow originates from the <body> section and points towards the text 'Idetação!!!' at the bottom left.

**Idetação!!!**

# Estrutura HTML – Dicas e Recomendações

- Proibido o “cruzamento” de *Tags*
  - `<head><body></head></body>`
- Utilizar letras minúsculas
- Utilizar aspas duplas nos atributos
- Colocar os principais atributos das *Tags*
- Identar o código ajuda o entendimento

# Estrutura HTML – *elementos*

- A recomendação completa do HTML
  - <https://html.spec.whatwg.org/multipage/>
- Tutorial da W3Schools:
  - <https://www.w3schools.com/html/>

# Estrutura HTML – *elementos*

- Elementos principais:

<code>&lt;html&gt; e &lt;/html&gt;</code>	Raiz do documento
<code>&lt;body&gt; e &lt;/body&gt;</code>	Corpo do documento
<code>&lt;head&gt; e &lt;/head&gt;</code>	Cabeçalho do documento
<code>&lt;title&gt; e &lt;/title&gt;</code>	Título do documento
<code>&lt;meta&gt;</code>	meta informação genérica

# Estrutura HTML – *elementos*

- Elementos de formulário:

<code>&lt;form&gt;</code> e <code>&lt;/form&gt;</code>	formulário interativo
<code>&lt;input&gt;</code>	Controle (campo) em formulário
<code>&lt;button&gt;</code> e <code>&lt;/button&gt;</code>	Botão
<code>&lt;label&gt;</code> e <code>&lt;/label&gt;</code>	Rótulo para campo de formulário
<code>&lt;textarea&gt;</code> e <code>&lt;/textarea&gt;</code>	campo de múltiplas linhas
...	



# Estrutura HTML – *elementos*

- Elementos de texto:

<code>&lt;p&gt;</code> e <code>&lt;/p&gt;</code>	parágrafo
<code>&lt;br&gt;</code> ou <code>&lt;br/&gt;</code>	Força uma quebra de linha
<code>&lt;strong&gt;</code> e <code>&lt;/strong&gt;</code>	Destaca o texto (negrito)
<code>&lt;em&gt;</code> e <code>&lt;/em&gt;</code>	Dá ênfase ao texto (itálico)
<code>&lt;a&gt;</code> e <code>&lt;/a&gt;</code>	âncoras e hyperlinks
<code>&lt;h1&gt;</code> e <code>&lt;/h1&gt;</code> , <code>&lt;h2&gt;</code> e <code>&lt;/h2&gt;</code> ...	Cabeçalho, apresenta o texto como títulos
<code>&lt;sub&gt;</code> e <code>&lt;/sub&gt;</code>	subscrito
<code>&lt;sup&gt;</code> e <code>&lt;/sup&gt;</code>	Sobrescrito
<code>&lt;del&gt;</code> e <code>&lt;/del&gt;</code>	Texto deletado
<code>&lt;ins&gt;</code> e <code>&lt;/ins&gt;</code>	Texto inserido
...	

# Estrutura HTML – *elementos*

- Elementos de tabela:

<code>&lt;table&gt;</code> e <code>&lt;/table&gt;</code>	tabela
<code>&lt;caption&gt;</code> e <code>&lt;/caption&gt;</code>	caption para tabela
<code>&lt;tr&gt;</code> e <code>&lt;/tr&gt;</code>	Linha da tabela
<code>&lt;th&gt;</code> e <code>&lt;/th&gt;</code>	Célula de cabeçalho da tabela
<code>&lt;td&gt;</code> e <code>&lt;/td&gt;</code>	Célula da tabela
<code>&lt;col&gt;</code> ou <code>&lt;col/&gt;</code>	Coluna na tabela
<code>&lt;colgroup&gt;</code> e <code>&lt;/colgroup&gt;</code>	Grupo de colunas na tabela
...	

# Estrutura HTML – *elementos*

- Elementos de listas:

<code>&lt;ol&gt;</code> e <code>&lt;/ol&gt;</code>	Lista ordenada
<code>&lt;ul&gt;</code> e <code>&lt;/ul&gt;</code>	Lista não ordenada
<code>&lt;li&gt;</code> e <code>&lt;/li&gt;</code>	Item de uma lista
...	

# Estrutura HTML – *elementos*

- Elementos diversos:

<code>&lt;div&gt; e &lt;/div&gt;</code>	elemento genérico /caixa para estilização
<code>&lt;img&gt;</code>	imagem
<code>&lt;hr&gt; ou &lt;hr/&gt;</code>	Insere uma Régua Horizontal (Horizontal Rule)
<code>&lt;span&gt; e &lt;/span&gt;</code>	elemento genérico /caixa para estilização
<code>&lt;style&gt; e &lt;/style&gt;</code>	informações de estilos
...	

# Estrutura HTML – *elementos*

- Alguns elementos em **desuso** ou **não são recomendados** segundo as Recomendações do W3C desde o HTML 4.01

<code>&lt;center&gt;</code> e <code>&lt;/center&gt;</code>	abreviação para DIV align=center
<code>&lt;font&gt;</code> e <code>&lt;/font&gt;</code>	estilização local para fonte
<code>&lt;strike&gt;</code> e <code>&lt;/strike&gt;</code> ou <code>&lt;s&gt;</code> e <code>&lt;/s&gt;</code>	coloca uma linha cortando um fragmento de texto
<code>&lt;u&gt;</code> e <code>&lt;/u&gt;</code>	Apresenta o texto sublinhado
...	

# Estrutura HTML – *elementos*

- Recomendações do W3C para o HTML 4 e para o HTML 5.x:
  - classificam os elementos **B** e **I** como “Elementos de estilização de fontes” e **desencorajam** o seu uso
  - classificam os elementos **STRONG** e **EM** como “Elementos de expressão” (ou “Elementos de frase”) e devem ser usados para **ênfatizar** uma palavra ou texto
  - São equivalentes visualmente
  - [https://www.w3schools.com/tags/tag\\_b.asp](https://www.w3schools.com/tags/tag_b.asp)
  - [https://www.w3schools.com/tags/tag\\_i.asp](https://www.w3schools.com/tags/tag_i.asp)

# Estrutura HTML – *Ligações*

- `<a>` e `</a>`
  - Faz uma ligação do texto para um outro endereço (link)
  - Principais atributos:
    - `href` □ contém o endereço (URL) a ser “linkado”
  - Exemplos:
    - `<a href="http://www.uol.com.br" target="_blank">clique aqui</a>`
    - `<a href="apresentacao.ppt">clique aqui</a>`

# Exemplo Documento HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Título da Página</title>
</head>
<body>

<h1>Meu Primeiro Cabeçalho</h1>
<p>Meu primeiro parágrafo.</p>
|
</body>
</html>
```

## Meu Primeiro Cabeçalho

Meu primeiro parágrafo.

- `<!DOCTYPE html>` declaração define que este documento é um documento HTML5
- `<html>` é o elemento raiz de uma página HTML
- `<head>` é o elemento que contém meta-informações sobre a página HTML
- `<title>` especifica um título para a página HTML (que é mostrado na barra de título do navegador ou na guia da página)
- `<body>` define o corpo do documento e é um recipiente para todos os conteúdos visíveis, tais como cabeçalhos, parágrafos, imagens, hiperlinks, tabelas, listas, etc.
- `<h1>` define um grande cabeçalho
- `<p>` define um parágrafo



# DTD

Document Type Definition

```
<!DOCTYPE html>
```

# Estrutura HTML – *DTD*

- HTML é baseada em SGML, então possui um **DTD** (**Document Type Definition**)
  - Define as regras para delinear a estrutura para uma dada classe de documentos
  - Um DTD ou uma referência para um DTD deve estar contido em qualquer documento conforme o padrão SGML
  - Acrônimo “DOCTYPE”

# Estrutura HTML 4.01 – *DTD*

- Existem três classes de HTML 4.01 diferentes:
  - HTML 4.01 Transitional
  - HTML 4.01 Strict
  - HTML 4.01 Frameset
- OBS: outras versões são divididas de maneira diferente

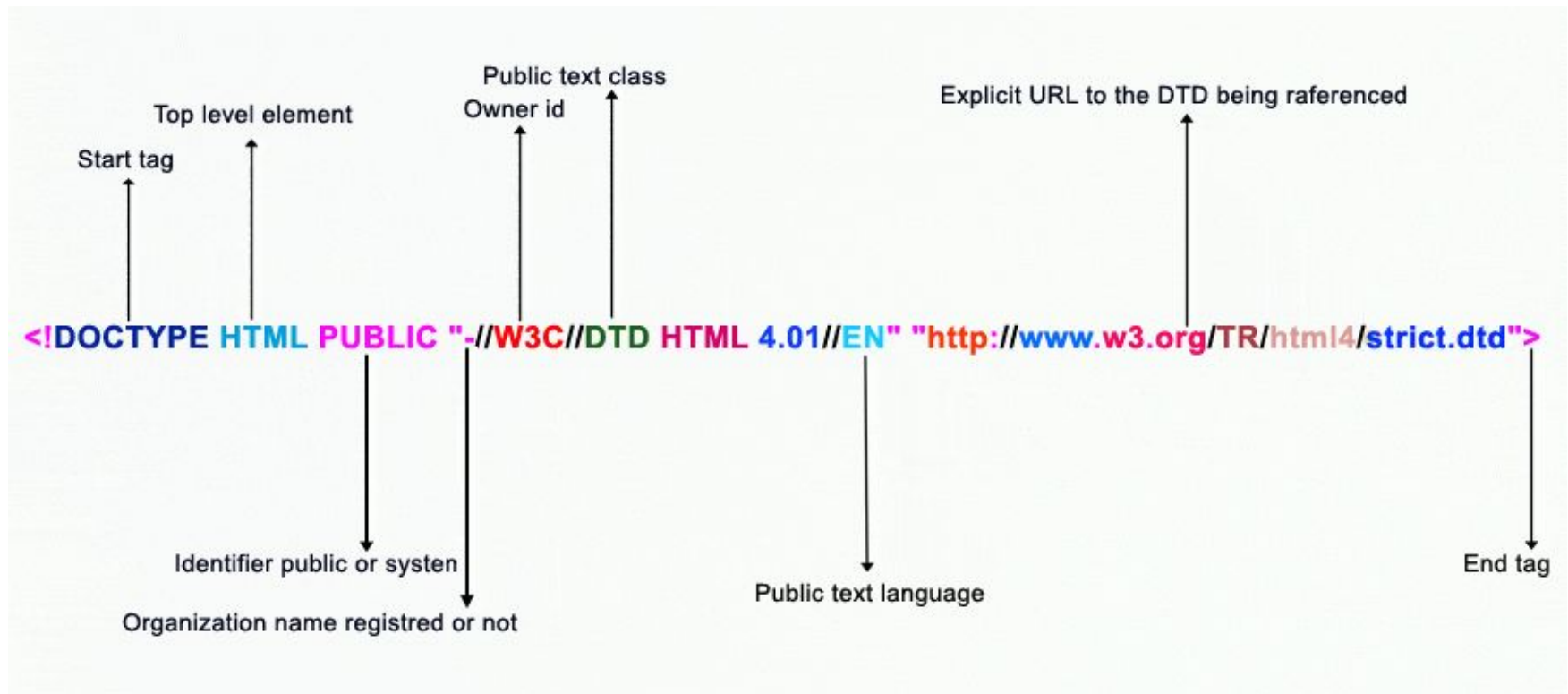
# Estrutura HTML 4.01 – *DTD*

- HTML 4.01 Transitional

- Deve ser usada na transição de HTML anterior à versão 4 para a versão 4.0
- Contém 11 elementos de apresentação e um conjunto de atributos já não utilizados
- Permite o uso de marcações que não são aceitas pelo grau “strict”
  - Exemplo: `<font size=12 face=Arial>Alo, pessoal!</font>`

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd" >
```

# Estrutura HTML 4.01 – *DTD*



# HTML 5 – *DTD*

- HTML 5
  - **Possui apenas uma declaração de DTD**
  - Além de única, ela é curta e fácil de lembrar
    - hoje em dia praticamente todos os desenvolvedores copiam e colam o longo e complicado doctype de algum lugar na hora de começar um novo documento HTML

```
<!DOCTYPE html>
```

# Estrutura HTML – *DTD*

- Os DTDs permitem a avaliação da programação em relação aos padrões web
  - De acordo com os graus citados:
  - "strict", "transitional" ou "frameset" para o HTML 4.01
  - Ou o único para o HTML 5

# Exemplo

- Exemplo de uma página HTML 5

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

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

```
    <title>Document Type Definition - DTD - HTML</title>
```

```
  </head>
```

```
  <body>
```

```
    Esta página baseia-se no HTML 5
```

```
  </body>
```

```
</html>
```



# Validador

- W3C Markup Validation Service
  - <http://validator.w3.org/>

# XHTML

Extensible HyperText Markup  
Language

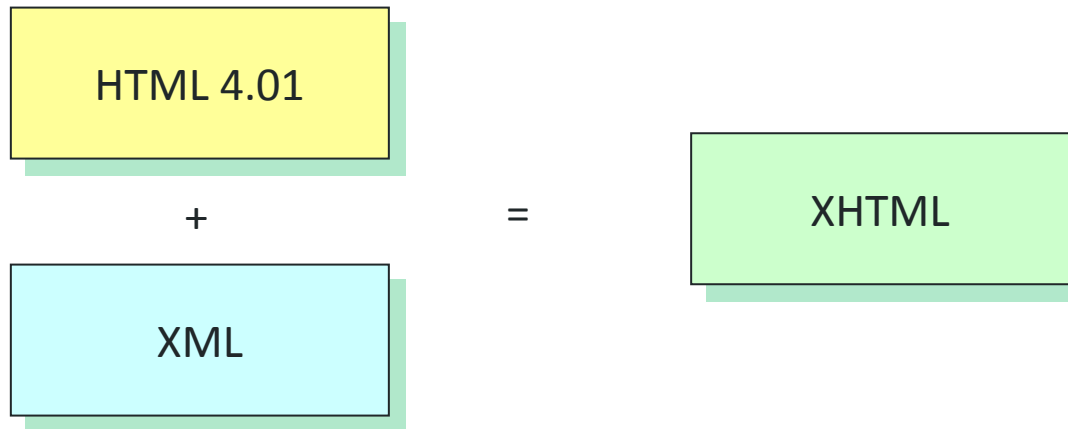
<html>  <xml>

# XHTML – *Introdução*

- Extensible HyperText Markup Language
  - *Linguagem de Marcação de Hipertexto Extensível*
- É um padrão W3C
  - Surgiu como uma necessidade de mais semântica
  - Quase idêntica ao HTML
  - Mais rigorosa que o HTML
  - XHTML é HTML definida como uma aplicação XML
  - XHTML é apoiada pela maior parte dos navegadores, incluindo TV digital, dispositivos embarcados
    - XHTML tinha como foco dispositivos que não rodavam HTML como PDAs, tablets, smartphones, mas hoje...

# XHTML – *Introdução*

- É uma reformulação da linguagem HTML baseada na XML
  - Adaptar as estruturas do HTML às regras do XML



# Versões Recentes

- HTML 4.0 – 1997
- HTML 4.01 - 1999
  - Limpeza do 4.0
- XHTML 1.0 - 2000
  - Similar 4.01 XML, no lugar de SGML
- XHTML 1.1 – 2001
  - Adeus aos frames
- XHTML 2.0 - 2010
  - <https://www.w3.org/TR/xhtml2/>

# XHTML - Hoje

- XHTML foi uma grande promessa, mas...
- Com o lançamento do HTML 5, o XHTML está entrando em desuso
- A W3schools não oferece mais o tutorial de XHTML
  - Há apenas uma página de comparação entre HTML e XHTML:  
[https://www.w3schools.com/html/html\\_xhtml.asp](https://www.w3schools.com/html/html_xhtml.asp)

# XHTML – *Regras Gerais*

1. Obrigação de elemento raiz
  - Todo documento deve possuir um elemento raiz no qual estarão contidos todos os outros elementos
  - No XHTML, o elemento raiz é o `<html>`

# XHTML – *Regras Gerais*

2. Nomes dos elementos e dos atributos devem ser escritos em minúsculo

**CORRETO:**

`<html>...</html>`

**INCORRETO:**

`<HTML>`, `<Html>`, etc.



# XHTML – *Regras Gerais*

3. Documento deve ser bem formado  
Fechamento correto e obrigatório das tags

*CORRETO: elementos aninhados.*

`<p>Aqui, um parágrafo em <strong>negrito</strong>.</p>`

*INCORRETO: elementos em "overlapping"*

`<p>Aqui, um parágrafo em<strong>negrito.</p></strong>`

# XHTML – *Regras Gerais*

4. Elementos e nomes de atributos devem ser em letra minúscula. XML é case-sensitive.

Exemplo: `<li>` and `<LI>` são tags diferentes

4. O uso de tags de fechamento é obrigatório

*CORRETO: elementos com fechamento*

`<a href="..." alt="..."> ...</a>  <br/>`

*INCORRETO: elementos sem fechamento*

` <br>`

# XHTML – *Regras Gerais*

6. Valores dos atributos devem estar entre aspas

*CORRETO: valores de atributos entre aspas*

```
<td rowspan="3">
```

*INCORRETO: valores de atributos sem o uso de aspas*

```
<td rowspan=3>
```

# XHTML – Regras Gerais

7. A sintaxe dos atributos deve ser escrita por completo

*CORRETO: atributos por extenso*

```
<input type="checkbox" name="caixa1" checked="checked">  
<dl compact="compact">
```

*INCORRETO: atributos abreviadamente*

```
<input type="checkbox" name="caixa1" checked>  
<dl compact>
```

# XHTML – *Regras Gerais*

## 8. Elementos vazios

*CORRETO: elementos vazios com terminação*

`<hr/> <br/>`

*INCORRETO: elementos vazios sem terminação*

`<hr> <br>`

# XHTML – *Regras Gerais*

## 9. Espaços em branco em valores de atributos

- eliminação dos espaços em branco
- mapeamento de sequências de espaços em branco (incluindo quebras de linha) e sua substituição por um espaço único entre palavras

# XHTML – *Regras Gerais*

## 10. Elementos script e style

São declarados como elementos com conteúdo #PCDATA

- < e & será tratado como uma marcação de início
- &lt; e &amp; serão reconhecidas pelos processadores XML como referências a entidades
- Evita o processamento destas entidades

```
<script type="text/javascript">  
  <![CDATA[  
    ... unescaped script content ...  
  ]]>  
</script>
```

# XHTML – Regras Gerais

## 11. Elementos vs Atributos ‘id’ e ‘name’

- HTML 4.01 usados como identificadores
- XML os identificadores são do tipo ID
- XHTML 1.0 o atributo id é definido como do tipo ID
  - Com o objetivo de certificar-se que os documentos XHTML 1.0 são documentos XML bem estruturados
  - Atributo name está formalmente em desuso e será excluído nas versões futuras de XHTML

*CORRETO: usando atributo id*

```

```

*INCORRETO: uso do atributo name*

```

```



# XHTML – *Regras Específicas*

- Declaração XML obrigatória

```
<?xml version="1.0" ?>
```

- Declaração de DTD obrigatória

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 2.0//EN"  
    "http://www.w3.org/MarkUp/DTD/xhtml12.dtd">
```

# XHTML – *Regras Específicas*

- Namespace e elemento raiz

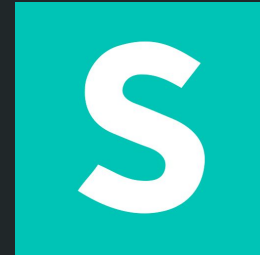
<html xmlns="http://www.w3.org/1999/xhtml">

<https://www.w3.org/TR/2006/WD-xhtml2-20060726/conformance.html>

- Elementos Obrigatórios

- <head>, <title> e <body>

# Semântica na Web



# Separar Conteúdo e Apresentação

- HTML não foi criado para efeitos de apresentação
  - mas os browsers acrescentaram elementos para esse efeito e os designers descobriram formas de manipular as tags existentes de forma a alterar o layout de uma página
- Com CSS já não é necessário incluir elementos de layout no HTML

# Separar Conteúdo e Apresentação

- Vantagens: Diminuir o tamanho dos arquivos, Flexibilidade, **Acessibilidade**, ...
- Não se trata apenas de retirar a apresentação do HTML, é necessário aprender a usar as marcas HTML no contexto certo: **A Semântica do HTML**

# Web Semântica

- **Semântica** = Estudo da linguagem do ponto de vista do significado das palavras
  - Do grego σημαντικός, derivado de **sema**, **sina**
  - Refere-se ao estudo do significado, em todos os sentidos do termo

# Web Semântica

- **Web Semântica** → uma web com toda sua informação organizada de forma que não somente seres humanos possam entendê-la, mas principalmente os computadores
  - A Web Semântica é uma evolução da nossa web atual
  - Com as informações devidamente organizadas, fica fácil de criar sistemas e robôs de busca mais inteligentes e ágeis
- **Devemos tentar sempre usar a marca mais apropriada a cada situação**

# Web Semântica

- A nossa web de hoje, é uma **web que apenas humanos entendem** as informações disponíveis. Com a Web Semântica, as máquinas compreenderão essas informações e assim, poderão nos auxiliar em tarefas corriqueiras, que antes eram feitas manualmente
- Atualmente é **extremamente complexo** fazer um sistema que leia e entenda de maneira sensata qualquer informação que a web provê



# Web Semântica

- A Web Semântica não será visualizada diretamente pelo browser
  - As tecnologias de Web Semântica podem agir por trás dos panos, resultando em uma melhor experiência do usuário, em vez de influenciar diretamente na aparência do browser

# Web Semântica

- Então, o ambiente de que estamos falando, terá informações devidamente identificáveis, que sistemas personalizados possam manipular, compartilhar e reusar de forma prática, as informações providas pela Web
  - Tente imaginar como o Google seria mais preciso em suas buscas se toda a informação da web estivesse organizada de uma maneira sensata
- Tudo isso soa meio estranho, futurístico?! Por onde podemos começar?

# Um pouco de Semântica na Web

- Título de um texto:

A. `<span class="title">Isto é um título</span>`

A. `<h1>Isto é um título</h1>`

A. `<p><b>Isto é um título</b></p>`

# Um pouco de Semântica na Web

- Título de um texto:

**`<h1>Isto é um título</h1>`**

- Sabemos que a marca `<h1>` funciona melhor em todos os browsers e dispositivos
- Descreve melhor o que se pretende
- Motores de pesquisa procuram marcas `<h1>` para descobrir o título de um documento

# Um pouco de Semântica na Web

- Uma lista:

```
<ul>
```

```
  <li>Item 1</li>
```

```
  <li>Item 2</li>
```

```
</ul>
```

```
- Item 1<br>
```

```
- Item 2<br>
```

# Um pouco de Semântica na Web

- Uma lista:

```
<ul>
```

```
  <li>Item 1</li>
```

```
  <li>Item 2</li>
```

```
</ul>
```

- Leitores para cegos anunciam listas
- Em alguns casos (exemplo: poema) a marca <br> pode ser aceitável

# Semântica — *Conclusão*

- Colocando semântica no HTML estaremos fazendo apenas o mínimo ainda!
- Mecanismos de busca
  - Busca por palavra-chave não é eficiente
  - Informação sem significado
  - Falta relevância na informação

# Conclusões

- HTML
  - Linguagem padrão para web
  - Necessita de nova postura do desenvolvedor
  - Padrões
- CSS
  - Facilita a manutenção do *layout* das páginas
  - Traz novos efeitos visuais ao HTML
  - Simples de utilizar
  - Separação de conteúdo e apresentação



# Conclusões

- Semântica no HTML
  - Temos que aprender a usar
  - Caos da web atual: deficiência no significado
- Tableless
  - Nova maneira de desenvolver HTML
  - Aprendizado
  - Postura

# HTML 5 – Melhorias na *Semântica*

Quais?

# HTML 5 – *Melhorias na Semântica*

- Vários novos elementos foram introduzidos no HTML 5 todos com a finalidade de facilitar a compreensão e a manutenção do código
  - Alguns são uma evolução natural do elemento <div> com foco na semântica
  - Outros surgiram da necessidade de padronizar a maneira de se publicar conteúdo, como acontece hoje com as imagens
  - Serão listados aqui, apenas os principais elementos

# HTML 5 – *Semântica na estrutura*

- **Semântica** -> um dos pontos mais importantes do **desenvolvimento com Padrões Web**
- Antes (HTML 4.01 e outras tecnologias):
  - Não é possível distinguir de forma automática as informações do “header” (cabeçalho) dos sites
  - Não é possível, de maneira automatizada, identificar o que é um rodapé ou a parte do layout que está exibindo um artigo
  - Não existe nenhum padrão de construção dos elementos para indicar áreas de conteúdo
  - A estrutura de um site não é óbvia para as máquinas
  - Não há semântica na organização

# HTML 5 - *Semântica na estrutura*

- Por exemplo, uma estrutura padrão:

```
<body>
  <div id="header">... </div>
  <div id="menu">...</div>
  <div class="post">...</div>
  <div id="sidebar">...</div>
  <div id="rodape">...</div>
</body>
```

- Seria substituída por:

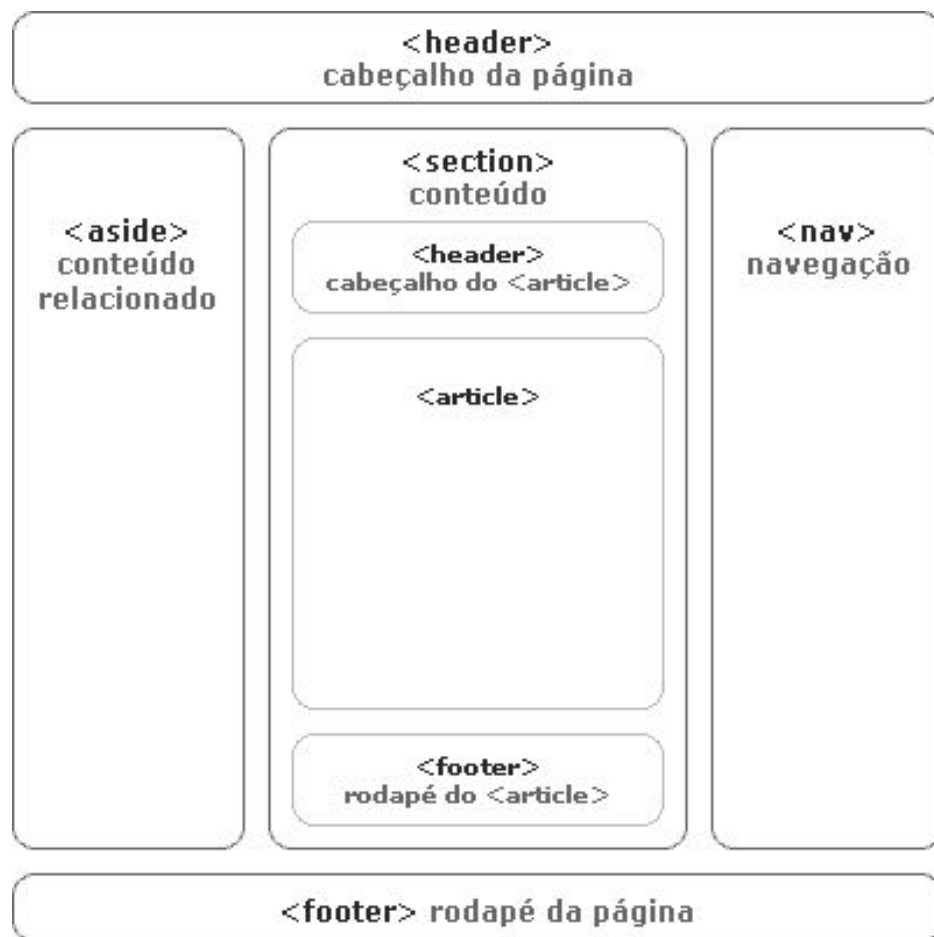
```
<body>
  <header>...</header>
  <nav>...</nav>
  <article>
    <section> ... </section>
  </article>
  <aside>...</aside>
  <footer>...</footer>
</body>
```

# HTML 5 - *Semântica na estrutura*

- Esses elementos são:
  - O elemento **header** define o cabeçalho
  - **Nav** define o menu ou a navegação do site
  - **Article** define uma parte da página que tem uma composição de formulários, textos etc.
  - O elemento **Section** define uma seção do layout em um determinado element
    - Pode conter um header e também um footer se preciso
  - O elemento **Aside** consiste em envolver informações que tem algo a ver com o conteúdo principal do site
    - Pode ser um menu lateral, um sidebar padrão com menu, banner, busca etc...
  - **Footer** define o rodapé do elemento ou do layout

# HTML 5 – Semântica na estrutura

- Elementos de estrutura



# HTML 5 – Semântica em novos elementos

- Elementos de conteúdo

**<figure>** - usado para associar uma legenda a uma imagem, vídeo, arquivo de áudio, objeto ou iframe:

```
<figure id="figura01">  
    
  <figcaption>Figura 1. Esquema de uma página em HTML 5.</figcaption>  
</figure>
```

**<canvas>** - através de uma API gráfica, irá renderizar imagens 2D dinâmicas que poderão ser usadas em jogos, gráficos, etc



# HTML 5 – Semântica em novos elementos

- Elementos de conteúdo

**<audio>** e **<video>** - usados para *streaming* (transmissão pela internet) de áudio e vídeo. É uma tentativa de criar um padrão em todos os navegadores como acontece hoje com as imagens:

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio tag.
</audio>
```

# HTML 5 – Semântica em novos elementos

- Elementos de conteúdo

**<time>** - representa data e/ou hora

```
<p> We open at <time>10:00</time> every morning. </p>  
<p>  
I have a date on <time datetime="2008-02-14">Valentines day</time>  
</p>
```

# HTML 5 – Semântica em novos elementos

- Elementos de conteúdo

<meter> - usada para representar medidas, que podem ser de distância, de armazenagem em disco, etc.

```
<meter value="2" min="0" max="10">2 out of 10</meter>  
<meter value="0.6">60%</meter>  
<meter value="0.6">Medium</meter>
```

# HTML 5 – Semântica em novos elementos

- **Elementos de conteúdo**

`<progress>` - Destina-se a marcar o andamento de uma tarefa

- O andamento pode ser tanto indeterminado quanto determinado, e, neste caso, indicado por um valor numérico entre 0 e um dado valor máximo, sendo assim possível saber com exatidão quanto falta para o término da tarefa

```
<progress value="3" max="6">Step 3 of 6</progress>  
<progress value="0.5">Half way!</progress>
```

# HTML 5

## Outras inovações

---

# HTML 5

- Foram feitas grandes alterações, que incluem:
  - novas APIs, entre elas uma para desenvolvimento de gráficos bidimensionais
  - controle embutido de conteúdo multimídia
  - aprimoramento do uso off-line
  - melhoria na depuração de erros
  - entre outros avanços

# HTML 5

- Esta evolução da linguagem padrão para web pode eliminar a necessidade de plug-ins para aplicações multimídia em navegadores

# HTML 5

- Internet passa por uma boa transformação
- tags de vídeo e suporte à tecnologia Canvas:
  - Opera
  - Google Chrome
  - Safari
  - Firefox
  - Edge/Internet Explorer



# HTML 5

- Com a evolução da linguagem, os navegadores passam da categoria “mostradores de páginas” para um renderizador de "web software"

# HTML 5 – Inovações

- Tags Canvas
  - Especializadas em renderizar imagens em bitmap, as tags canvas serão específicas para a edição breve de imagens através de APIs ou JavaScript
  - Esse tipo de edição acontece pura e exclusivamente de maneira muito similar a outros geradores de imagem em duas dimensões (2D)
  - Canvas Tags são compatíveis com CSS



# HTML 5 – Inovações

- Tags de vídeo
  - Incluir vídeos em HTML como conhecemos hoje, utilizando códigos para o “embed”, ou seja, incorporar vídeos à página será muito mais simples.
  - O HTML 5 possui tags específicas para a inserção de vídeos no corpo da página.
  - Para os vídeos a situação será muito semelhante ao que é feitos hoje com as imagens.



# HTML 5 – Inovações

- Geolocalização

- O HTML 5 também possui maneiras de descobrir a sua localização e informá-la aos sites e serviços que você acessa
- As APIs serão o ponto forte para determinar a localização de um usuário. Por isso, este recurso de geolocalização permite o que se chama de “geotagging”
- Pode-se fornecer ao usuário conteúdos específicos para o local em que ele está



# HTML 5 – Inovações

- Caching de aplicações

- Com o HTML 5 os aplicativos web poderão ser acessados offline via cache
- De acordo com a publicação provisória da W3C, o arquivamento desses aplicativos será feito via URL em que cada uma delas possui uma categoria diferente
  - As entradas mestre são aqueles documentos que foram adicionados ao cache por um contexto de navegação indicado por um atributo de manifesto
  - Já o manifesto é a fonte da URL indicada na Entrada mestre do HTML. Esses arquivamentos ainda podem ser agrupados ou não



# HTML 5 – *Inovações*

- Base de dados
  - No HTML 5, algumas vantagens serão implementadas como a possibilidade de entradas de valores ou palavras chave, além do banco de dados SQL



# HTML 5 – Inovações



- E o que muda para o usuário?
  - O usuário sentirá boas diferenças ao navegar pela Internet
    - HTML 5 irá facilitar muito a vida de todos
  - Para os desenvolvedores, fica mais rápido produzir sites cada vez mais fáceis e limpos para que os usuários possam aproveitar ao máximo cada uma dessas melhorias

# HTML 5 – *Elementos retirados*

- Alguns elementos **não** existirão mais no HTML 5
  - Alguns foram retirados porque sua função é puramente visual, e devem ser substituídos por uma declaração no CSS, como:
    - `<basefont>`, `<big>`, `<center>`, `<font>`, `<s>`, `<strike>`, `<tt>` e `<u>`
  - Outros foram retirados porque afetam negativamente a acessibilidade do site:
    - `<frame>`, `<frameset>` e `<noframes>`
  - Apesar de serem considerados antigos, `<b>` e `<i>` ainda serão reconhecidos e renderizados para fins de formatação, mas devem ser substituídos sempre que possível pelos elementos `<strong>` e `<em>`, respectivamente



# HTML 5 – *Elementos retirados*

- Também foram retirados alguns atributos, seja porque caíram em desuso ou porque podem ser substituídos semanticamente por declarações no CSS para definir o visual dos elementos.
- Os principais atributos retirados são:
  - target no elemento <a>;
  - align nos elementos <table> e demais tags de tabelas, <iframe>, <img>, <input>, <hr>, <div>, <p>, entre outros
  - background em <body>
  - bgcolor nos elementos de tabela e no <body>
  - border em <table> e <object>
  - cellpadding e cellspacing em <table>
  - height em <td> e <th>
  - width nos elementos <hr>, <table>, <td>, <th> e <pre>
  - hspace e vspace em <img> e <object>
  - noshade e size em <hr>

# Links interessantes

- Os Navegadores já implementam HTML 5???  
**Não completamente !!!**
- Avaliador de navegador:
  - <http://html5test.com/>

# Links interessantes

- Exemplos:

- <http://html5-showcase.com/>
- <http://html5demos.com/>
- <http://html5sites.com/>
- <http://html5gallery.com.br/>

# CSS

## *Cascading Style Sheets*

```
div.direita {  
    float: right;  
    color: white;  
    background-color: black;  
}
```

# Introdução

- CSS = *Cascading Style Sheets*
- CSS surgiu para **facilitar a manutenção visual** (*layout*) dos hiperdocumentos
- Permite que diversas **classes** de *layout* sejam definidas e reutilizadas
- Diferença entre os *browsers*

# Estrutura do CSS

- Estrutura completa

Definição da classe

```
{  
    Definição de elemento  
    Definição de elemento  
    ...  
}
```

# Estrutura do CSS

- **Definição da classe**

- `[nome da tag][.nome classe][:extensão]`
- `[]` - opcional (tag ou classe deve existir)

- **Exemplos:**

- `a.verde: hover`
- `.verde`
- `label`
- `input.verde`
- `#id`

# Estrutura do CSS

- **Definição do elemento**

- nome do elemento: valor;

- **Exemplos:**

- `font-family: arial, verdana, times;`
- `color: #FFFFFF;`
- `text-decoration: underline;`



# Estrutura do CSS

- Exemplo de uma definição CSS:

```
.texto {  
    font-family: arial, verdana, times;  
    font-size: 11px;  
    color: #FFFFFF;  
    text-decoration: none;  
}
```

# Integração do CSS no HTML

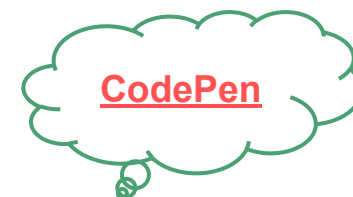
- Integração CSS e HTML é feita através do atributo **class**
- Exemplo:
  - `<a class="verde" href="http://www.facom.ufms.br">`  
clique nesta âncora para abrir o site da Facom  
`</a>`
  - `<td class="lista">Lista de alunos</td>`

# Integração do CSS no HTML

- Tag específica para aplicação de CSS
- `<span>...</span>`
  - Substitui o `<font>...</font>`
  - Atributo class
  - [https://www.w3schools.com/tags/tag\\_span.asp](https://www.w3schools.com/tags/tag_span.asp)
- Exemplo:
  - **`<span class="verde">`**  
Esse texto utilizará as definições da classe CSS de nome verde  
**`</span>`**

# Integração do CSS no HTML

- Definição na mesma página HTML
- `<style>` e `</style>`
  - Indica que definições de estilo virão entre as *Tags*
  - Atributo: `type="text/css"`
  - [https://www.w3schools.com/tags/tag\\_style.asp](https://www.w3schools.com/tags/tag_style.asp)



```
<!DOCTYPE html>
```

```
<!-- Exemplo de CSS com definições na mesma página -->
```

```
<html>
```

```
  <head>
```

```
    <title>Exemplo de CSS</title>
```

```
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

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

```
      .texto
```

```
      {
```

```
        color: #FF0000;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <p><span class="texto">Texto que sofrerá alterações de CSS</span>, mas no  
    mesmo parágrafo, esse texto aqui terá o estilo padrão</p>
```


```
  </body>
```

```
</html>
```

Definição do estilo

Uso da classe de estilo

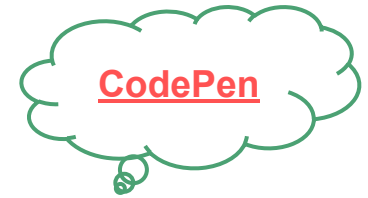
# Integração do CSS no HTML

- Definido em um arquivo próprio
  - Montar arquivo só com as definições (sem as tags `<style>` e `</style>`)  

  - Incluir tal arquivo através da seguinte tag:

**`<link href="nomedoarquivo" rel="stylesheet" type="text/css">`**

[https://www.w3schools.com/tags/tag\\_link.asp](https://www.w3schools.com/tags/tag_link.asp)

## ARQUIVO exemplo.html



```
<!DOCTYPE html>
<!-- Exemplo de CSS com definições em arquivos diferentes -->
<html>
  <head>
    <title>Exemplo de CSS</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <link href="classes.css" rel="stylesheet" type="text/css">
  </head>
  <body>
    <span class="texto">Texto que sofrerá alterações de CSS</span>
  </body>
</html>
```

## ARQUIVO classes.css

```
.texto {
  color: #FF0000;
}
```

# Estrutura do CSS

- .texto
  - Pode ser utilizado em todas as *tags* que suportam CSS
- A.texto
  - Só pode ser utilizado em *tags* A
- A
  - Define um padrão para todas as *tags* A sem classe



# Principais usos do CSS

- Definir formatação de textos
- Definir links
- Definir estilos de caixas de formulários

# Exemplos

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
  <head>
    <title>Exemplo de CSS</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <style type="text/css">
      .texto
      {
        font-family: arial;
        font-size: 16px;
        text-decoration: none;
        font-style: italic;
        color: #FFFF00;
      }
    </style>
  </head>
  <body>
    <div class="texto">Texto que sofrerá alterações de CSS</div >
  </body>
</html>
```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
  <head>
    <title>Exemplo de CSS</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <style type="text/css">
      a.especial:visited
      {
        color: #000000;
        font-style: italic;
        text-decoration: none;
      }
      a.especial:hover
      {
        color: #00FF00;
        font-style: italic;
        text-decoration: underline;
      }
      a.especial:link
      {
        color: #FF0000;
        font-style: italic;
        text-decoration: none;
      }
    </style>
  </head>
  <body>
    <p><a class="especial" href="http://www.ufms.br">
      âncora para o site da UFMS</a></p>
  </body>
</html>

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
  <head>
    <title>Exemplo de CSS</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <style type="text/css">
      .texto{
        font-family: verdana;
        font-size: 11px;
        font-weight: bold;
        text-decoration: underline;
        font-style: italic;
        color: #FF0000;
      }
      .caixa{
        background : #FF9900;
        border-left : thin solid #444444;
        border-bottom : medium solid #444444;
        border-right : medium solid #444444;
        border-top : thick solid #444444;
        font-weight: bold;
        font-family: verdana, helvetica;
        font-size: 12px;
        color: #0000FF;
      }
    </style>
  </head>
  <body>
    <form action="tratardados.php" name="form" method="post">
      <p> <label class="texto">Digite seu RG: </label>
        <input class="caixa" type="text" name="rg" size="10"
          maxlength="10">
      </p>
    </form>
  </body>
</html>

```

# Dicas e Recomendações

- Definir CSS em arquivos próprios
  - Facilita muito a manutenção
- Testar sempre em diversos *browsers*
- Evitar definir formatações específicas
  - Dificultará manutenção
  - `<font color="#FFFFFF">`

# Dicas e Recomendações

- Utilizar classes CSS genéricas
  - .texto .verde .destaque .titulo
- Utilizar nomes fáceis de lembrar nas classes
- Pesquisar novos usos na Internet
  - Existem muitos elementos!

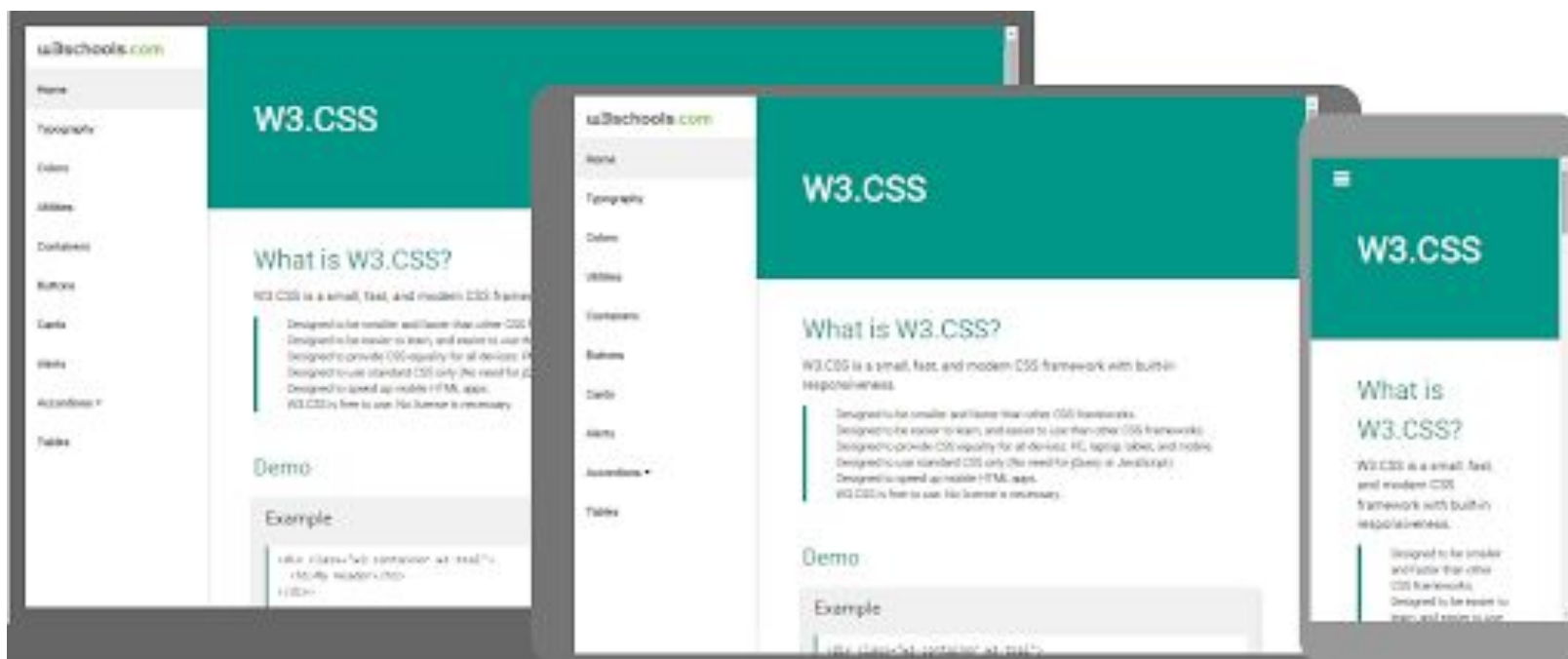
# Exemplo de CSS

- <http://www.csszengarden.com/tr/portuguese/>



# Dicas de CSS

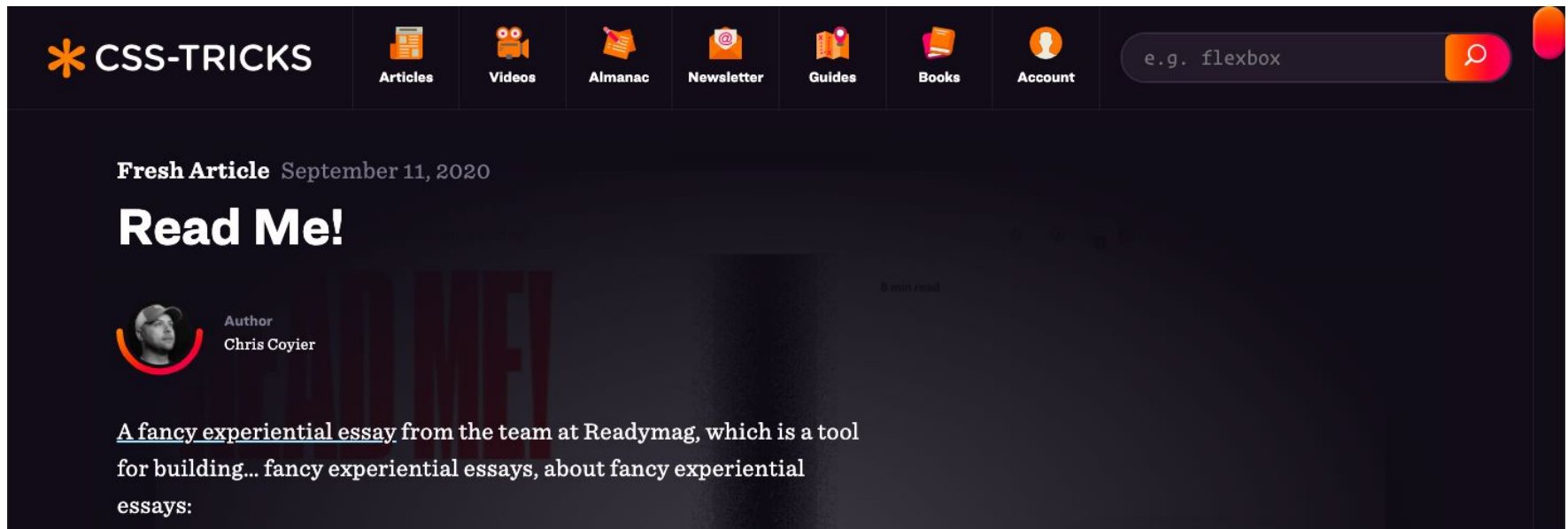
- <https://www.w3schools.com/css/default.asp>





# Dicas de CSS

- <https://css-tricks.com>



**Design**

**Tableless**



# Design Tableless

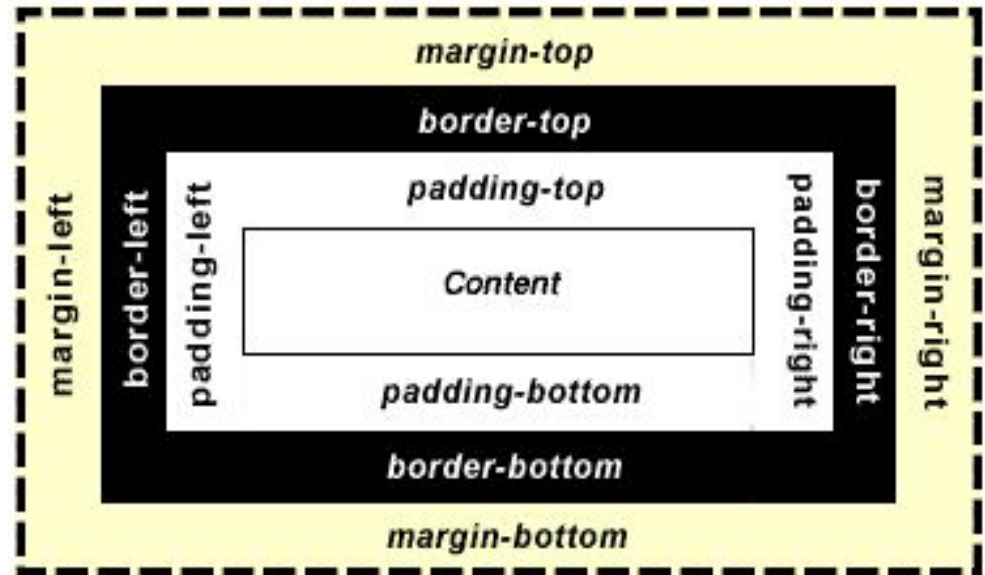
- Idéia principal usar DIVs no lugar de <table>
- Usar <table> **somente quando a informação for tabular**
- Separar estrutura do conteúdo da página
- DIVs funcionam como uma caixa
- Usar CSS para posicionar os DIV

# Design Tableless

- Como planejar uma página tableless
  - Dividir o layout em seções:
    - Menu, Conteúdo, Procura, Cabeçalho, etc
  - Utilizar divs para representar estas seções

# DIVS

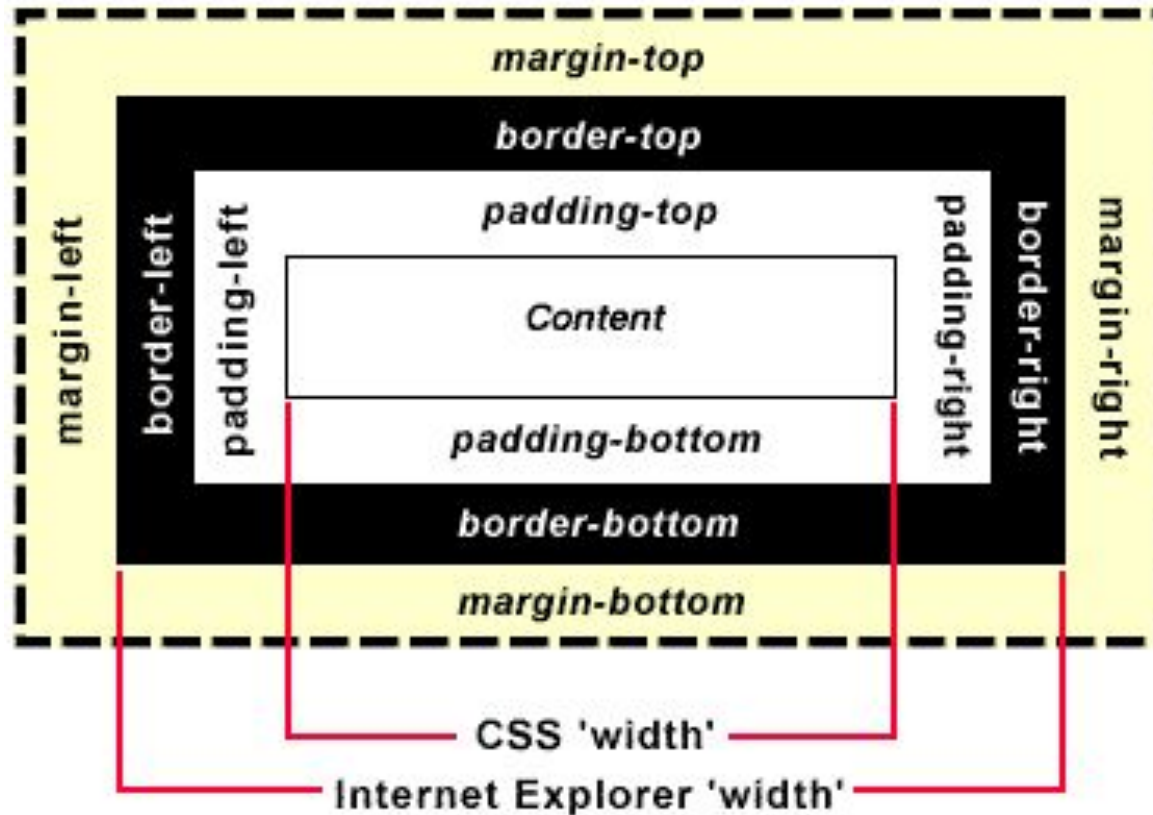
- Cada div possui
  - Conteúdo
    - Dado a ser exibido, texto e imagens.
  - Padding
    - Espaço em branco ao redor do conteúdo
  - Border
    - Definida depois do padding
  - Margin
    - Definida depois do Border



# Tamanho do DIV

- Web Standard (Firefox, Chrome, Opera)
  - Width é a área do conteúdo
  - Padding, border margin são espaços calculados a parte
  - **Para criar um div de 100px**
    - **width: 100px; padding: 10px; border: 10px**
- IE
  - Width é o tamanho do div todo inclui todos os tipos de margem: Padding, border
  - **Para criar um div de 100 px**
    - **width: 140px; padding: 10px; border: 10px**

# Tamanho do Div



# Exemplo DIV

```
<style type="text/css">
  .article1 {
    width: 250px;
    padding: 5px;
    float: right;
    font-family: verdana, arial, sans-serif;
  }

  .article2 {
    width: 200px;
    padding: 5px;
    float: left;
  }

  .red {
    background-color: red;
  }
</style>
```

```
<body>
  <div class="article1">
    Previsao do Tempo <br>
    Campinas – Muita chuva
  </div>

  <div class="article2 red">
    Cotacao de modedas <br>
    Dolar – R$ 2,15
  </div>
</body>
```



# Controlando o comportamento de Divs

- Modelos de comportamento
  - **inline**
    - Usa somente o espaço necessário
    - Elementos são posicionados lado a lado
  - **block**
    - Usam 100% da largura do container
    - Novos blocos são adicionados abaixo do outro
  - **table**
    - Simula o comportamento da tag <table>
  - **flex**
    - Uma forma mais eficiente para posicionar e distribuir espaços entre os elementos
  - **grid**
    - Um sistema de posicionamento bidimensional de elementos
  - **none**
    - O elemento é completamente removido

# Posicionamento dos Divs

- Modelos de Posicionamento
  - **static**
    - (default) – coloca os divs em um fluxo contínuo que representa a página HTML
    - Similar ao layout default do HTML
  - **relative**
    - Coloca o div uma posição relativa a outro div
    - Apenas o div atual é afetado, os demais divs são posicionados de acordo com o modo default estático
  - **absolute**
    - Coloca o div em uma posição fixa absoluta na página (x, y, z)
    - Não afeta outros elementos colocados acima ou abaixo do div
  - **fixed**
    - Similar ao modo *absolute*, mas é fixo absolutamente a janela
    - Ao dar scrolls na página, o div não se moverá

# Links

- Para saber mais:
  - <http://www.w3.org/Style/CSS/>
  - <http://www.tableless.com.br>
  - <https://css-tricks.com>

# Referências

- W3C – Escritório Brasil
  - <http://www.w3c.br/>
- <http://www.tableless.com.br/html5-estrutura-semantic>
- <http://www.w3schools.com/html5>

# Referências

- [www.w3c.org](http://www.w3c.org)
- Cover Pages: Extensible Markup Language (XML).  
<http://xml.coverpages.org/xml.html>
- The XML FAQ v4.0 (2005-01-01) <http://www.ucc.ie/xml/>
- Harold, E. R. (2001) XML Bible. Second Edition, Hungry Minds
- <http://www.w3.org/TR/xhtml11/>