

HTML & CSS

Informática / LECA

Marco Costa

Departamento de Ciência de Computadores
Faculdade de Ciências, Universidade do Porto

2005 / 06



Conteúdo

1 HTML

- Introdução
- Páginas básicas
- Texto
- Listas
- Imagens
- Links
- Tabelas
- Outras tags

2 CSS

- Introdução
- Selecção de elementos
- Propriedades de formatação

3 Bibliografia

1 HTML

- Introdução
- Páginas básicas
- Texto
- Listas
- Imagens
- Links
- Tabelas
- Outras tags

2 CSS

- Introdução
- Selecção de elementos
- Propriedades de formatação

3 Bibliografia

1 HTML

- Introdução
- Páginas básicas
- Texto
- Listas
- Imagens
- Links
- Tabelas
- Outras tags

2 CSS

- Introdução
- Selecção de elementos
- Propriedades de formatação

3 Bibliografia

HTML

HTML = HyperText Markup Language

- **Hyper** porque não é linear (pode-se navegar entre documentos HTML seguindo links)
- **Text** porque um documento HTML é um ficheiro de texto
- **Markup** porque usam-se “marcas” ou “etiquetas” no texto para indicar parágrafos, títulos, listas, etc.
- **Language** porque é uma linguagem

Ou seja...

- **HTML** é a linguagem utilizada para criar as “páginas” da Web
- Uma página corresponde a um ficheiro de texto, que pode referenciar outros ficheiros (por exemplo, imagens)
- Os ficheiros HTML têm a extensão **.html** ou **.htm**

Tags

Definição

- Um documento HTML é formado por uma hierarquia de elementos (isto é, elementos que contêm outros)
- A cada tipo de elemento corresponde uma **tag**
- Por exemplo, aos parágrafos correspondem a tag **p**
- Para iniciar um parágrafo escreve-se **<p>**, e para finalizá-lo, **</p>** (note-se o símbolo **/**)
- Exemplo de um parágrafo: **<p>Isto é um parágrafo</p>**
- As tags podem conter atributos
- Exemplo de uma tag com dois atributos:
<p class="topo" id="nada">Isto é um parágrafo</p>
- As tags que não contêm nada entre a abertura e o fecho podem ser abreviadas (por exemplo, **
** em vez de **
</br>**)

Browsers

O que é um browser?

Um browser é um programa utilizado para navegar na Web, isto é, visualizar páginas e navegar entre elas

Exemplos de browsers

- Internet Explorer
- Mozilla Firefox
- Konqueror
- Safari
- Opera

Servidores Web

O que é um servidor Web?

- Um servidor Web é um programa que disponibiliza um conjunto de páginas Web
- Um dos servidores Web mais utilizados é o **Apache**
- A comunicação entre browsers e servidores é feita com base nas regras especificadas no protocolo **HTTP** (HyperText Transfer Protocol)

Endereços de páginas Web

Definição e exemplo

- O endereço de uma página Web assume a forma de um URL (*Uniform Resource Locator*)
- Estes URL's podem ser divididos em três partes: o protocolo, o endereço do servidor, e o endereço da página dentro do servidor
- Por exemplo, o endereço `http://www.dcc.fc.up.pt/~francisco` indica o protocolo **HTTP**, o servidor `www.dcc.fc.up.pt`, e a página `/~francisco` nesse servidor

Modelo cliente servidor

Explicação

A Web funciona com base no modelo cliente-servidor: os browsers são os clientes (pedem as páginas) e são servidos pelos servidores Web (fornecem as páginas)

Exemplo

- O João inicia o Internet Explorer e coloca o endereço `http://www.dcc.fc.up.pt/~francisco`
- O browser contacta o servidor no endereço `www.dcc.fc.up.pt` e pede-lhe a página `/~francisco`
- O servidor localiza a página e envia-a ao browser
- O browser recebe a página, interpreta-a e mostra-a ao João
- O João selecciona um dos links da página
- O browser contacta o servidor indicado no link, ...

História da Web

Cronologia

- Em 6 de Agosto de 1991, Tim Berners-Lee, o criador da Web, disponibilizou as primeiras páginas Web
- Em Setembro de 1993 foi criado o browser **Mosaic**, que contribuiu para aumentar a popularidade da Web
- Em 1993 foi criado o primeiro *search engine*, o Lycos
- Em 1994 foi fundado o **World Wide Web Consortium** (W3C), uma consórcio que cria normas e recomendações relacionadas com a Web
- Em 1994 foi lançada a primeira versão do browser **Netscape**
- Em 1995 a Microsoft lançou o Internet Explorer
- Em 2002 foi lançada a primeira versão do Mozilla Firefox, um dos descendentes do Netscape
- Em 2005, a Yahoo anunciou que o seu *search engine* indexa mais de 20 mil milhões de páginas e imagens

1 HTML

- Introdução
- **Páginas básicas**
- Texto
- Listas
- Imagens
- Links
- Tabelas
- Outras tags

2 CSS

- Introdução
- Selecção de elementos
- Propriedades de formatação

3 Bibliografia

Exemplo de uma página básica

Texto do ficheiro HTML

```
<html>
  <head><title>Hello World</title></head>
  <body>
    <p>Hello world</p>
  </body>
</html>
```

Elementos/tags que compõem o documento HTML

- A tag **html** indica um documento HTML; abre e fecha o documento
- A tag **head** indica o cabeçalho do documento
- A tag **title** indica o título do documento, e deve aparecer na **head**
- A tag **body** indica o corpo do documento
- A tag **p** indica um parágrafo (o texto aparece entre **<p>** e **</p>**)

Algumas complicações — Versões

DTD's

- Desde que foi criado, o HTML sofreu várias alterações
- Por isso, existem várias versões do HTML
- Cada versão é definida por um DTD (*Document Type Definition*)
- Vamos utilizar a versão **XHTML 1.0 Transitional**
- Devemos indicar no próprio documento HTML qual a versão utilizada
- Antes do elemento **html** escrevemos:

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

- E acrescentamos o atributo **xmlns** ao elemento **html**, ficando:

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

Algumas complicações — Codificação de caracteres

Codificação de caracteres

- Os documentos HTML podem ser escritos utilizando diferentes codificações de caracteres
- Devemos indicar no documento qual a codificação utilizada
- Incluímos um elemento `meta` no cabeçalho do documento
- Se utilizarmos a codificação `iso-8859-1`, escrevemos:

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

Algumas complicações — Resultado

Versão final do documento apresentado anteriormente

```
<!DOCTYPE html PUBLIC
    "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Hello World</title>
    <meta http-equiv="Content-Type"
        content="text/html; charset=iso-8859-1"/>
  </head>
  <body>
    <p>Hello world</p>
  </body>
</html>
```

Dica

Utilize este documento HTML como ponto de partida (**copy-n-paste**)

Validação de documentos HTML

O que é? Como se faz?

- Os browsers não se costumam queixar dos erros contidos nos documentos HTML
- Por isso, os erros passam muitas vezes despercebidos
- Mas existem formas de validar os documentos
- Pode-se assim detectar a presença de erros e, mais importante, corrigi-los
- Por exemplo, utilizando o validador *online* disponível no seguinte endereço:

<http://validator.w3.org/>

1 HTML

- Introdução
- Páginas básicas
- **Texto**
- Listas
- Imagens
- Links
- Tabelas
- Outras tags

2 CSS

- Introdução
- Selecção de elementos
- Propriedades de formatação

3 Bibliografia

Parágrafos

Funcionamento

- A tag `p` define um parágrafo
- A tag `br` define uma quebra de linha (o parágrafo actual continua, mas na linha seguinte)

Notas

- 1 A tag `br` é normalmente utilizado na forma abreviada
- 2 Não interessa quantos espaços se escreve entre as palavras; só é considerado um deles

Parágrafos — Exemplo

Exemplo

```
<p>Este é o primeiro parágrafo.</p>  
<p>Este é o segundo parágrafo. <br/> Isto também <br/>  
pertence ao <br/> segundo parágrafo.</p>
```

Este é o primeiro parágrafo. Este é o primeiro parágrafo.

Este é o segundo parágrafo.

Isto também
pertence ao
segundo parágrafo.

Títulos

Funcionamento

- As tags **h1**, **h2** e **h3** (e outras) definem títulos
- **h1** é um título mais importante que **h2**, que é mais importante que **h3**

Títulos — Exemplo

Exemplo

```
<h1>Introdução</h1>  
<p>Isto é a introdução.</p>  
<h2>Objectivos</h2>  
<p>Aqui vou falar dos  
    objectivos.</p>  
<h3>Objectivo A</h3>  
<p>O objectivo A é ...</p>
```

Introdução

Isto é a introdução.

Objectivos

Aqui vou falar dos objectivos.

Objectivo A

O objectivo A é ...

Formatação de texto

Funcionamento

- A tag **b** define texto em “negrito”
- A tag *i* define texto em itálico
- As tags ^{sup} e _{sub} definem texto superior e inferior à linha

Exemplo

```
<p><b>Negrito</b>, <i>itálico</i>,  
linha<sub>inferior</sub>, linha<sup>superior</sup></p>
```

Negrito, *itálico*, linha_{inferior}, linha^{superior}

1 HTML

- Introdução
- Páginas básicas
- Texto
- **Listas**
- Imagens
- Links
- Tabelas
- Outras tags

2 CSS

- Introdução
- Selecção de elementos
- Propriedades de formatação

3 Bibliografia

Listas

Funcionamento

- A tag `ul` define uma lista não numerada
- A tag `ol` define uma lista numerada
- A tag `li` define um elemento de uma lista; deve ser incluído dentro de uma lista (numerada ou não)

Listas — Exemplo

Lista não numerada

```
<p>Lista não numerada</p>  
<ul>  
  <li>Um</li>  
  <li>Dois</li>  
  <li>Três</li>  
</ul>
```

Lista não numerada

- Um
- Dois
- Três

Lista numerada

```
<p>Lista numerada</p>  
<ol>  
  <li>Um</li>  
  <li>Dois</li>  
  <li>Três</li>  
</ol>
```

Lista numerada

1. Um
2. Dois
3. Três

Listas multi-nível

Funcionamento

- Listas multi-nível são “listas dentro de listas”
- Para criar uma lista dentro de outra lista, abre-se uma nova lista dentro de um elemento da lista anterior
- Podem-se misturar listas ordenadas com listas não ordenadas

Exemplo

```
<p>Lista com dois níveis</p>  
<ul>  
  <li>Nível um  
    <ul>  
      <li>Nível dois</li>  
    </ul></li>  
  <li>Nível um</li>  
</ul>
```

Lista com dois níveis

- Nível um
 - Nível dois
- Nível um

1 HTML

- Introdução
- Páginas básicas
- Texto
- Listas
- **Imagens**
- Links
- Tabelas
- Outras tags

2 CSS

- Introdução
- Selecção de elementos
- Propriedades de formatação

3 Bibliografia

Imagens

Funcionamento

- Para incluir uma imagem num documento HTML utiliza-se a tag `img`
- O atributo `src` é utilizado para indicar a localização da imagem, na forma de um URL
- O atributo `alt` deve ser utilizado para indicar uma descrição curta da imagem
- Essas descrições são utilizadas em alternativa à imagem, por exemplo, quando a página é ouvida em vez de lida
- Os atributos `width` e `height` podem ser utilizados para indicar as dimensões da imagem (permitindo redimensionar a imagem)
- A tag `img` é normalmente utilizada na forma abreviada

Imagens — Exemplo

Imagem local

```

```



Imagens — Exemplo

Imagem (remota e redimensionada)

```
<p>Bandeira de Portugal</p>
```

```

```

Bandeira de Portugal



1 HTML

- Introdução
- Páginas básicas
- Texto
- Listas
- Imagens
- **Links**
- Tabelas
- Outras tags

2 CSS

- Introdução
- Selecção de elementos
- Propriedades de formatação

3 Bibliografia

Links

Funcionamento

- Um link, quando activado pelo utilizador, permite-lhe “navegar” para outra localização (normalmente, outra página)
- Para criar um link utiliza-se a tag `a`, indicando-se no atributo `href` o URL destino
- O texto (ou outro conteúdo) que representa o link é colocado dentro da tag `a` (entre `<a>` e ``)

Exemplo

```
<p>A <a href="http://www.fc.up.pt/">FCUP</a> pertence  
à <a href="http://www.up.pt/">UP</a>.</p>
```

A FCUP pertence à UP.

Links locais

Funcionamento

- Um link local é um link que aponta para uma localização na página actual
- Quando o utilizador activa um link local, o browser avança ou recua (faz *scroll*) na página actual até que o local para onde o link aponta fique visível
- A tag `a` é utilizada para criar os links e também para criar ancoras (sítios para onde os links locais podem apontar)
- Para criar uma ancora utiliza-se a tag `a`, indicando-se com o atributo `name` o nome da ancora
- Para apontar para uma ancora utiliza-se também a tag `a`, indicando-se com o atributo `href` o nome da ancora para onde o link aponta (coloca-se `#` antes do nome)

Links locais — Exemplo

Exemplo

```
<p>  
Link para o <a href="#segundo">segundo parágrafo</a>.  
</p>  
<p><a name="segundo"/>Segundo parágrafo.</p>
```

Link para o [segundo parágrafo](#).

Segundo parágrafo.

Links com imagens

Funcionamento

- As imagens podem ser utilizadas para representar links
- Em vez de texto, coloca-se uma imagem entre `<a>` e ``
- Nestas situações um link é activado quando a imagem é clicada

Exemplo

```
<p>A <a href="http://www.fc.up.pt/">  
  
</a> pertence à <a href="http://www.up.pt/">UP</a>.</p>
```



1 HTML

- Introdução
- Páginas básicas
- Texto
- Listas
- Imagens
- Links
- **Tabelas**
- Outras tags

2 CSS

- Introdução
- Selecção de elementos
- Propriedades de formatação

3 Bibliografia

Tabelas

Funcionamento

- As tabelas permitem organizar conteúdo de forma tabular
- A tag **table** cria uma tabela
- A tag **tr** cria uma linha numa tabela (dentro de **table**)
- A tag **td** cria uma célula (dentro de **tr**)
- A tag **th** cria uma célula “título”, cujo texto é normalmente mostrado centrado e a negrito (dentro de **tr**)

Exemplo

```
<table>
<tr><th>Nome</th><th>Idade</th></tr>
<tr><td>João</td><td>25</td></tr>
<tr><td>Maria</td><td>47</td></tr>
</table>
```

Nome Idade	
João	25
Maria	47

Tabelas com limites

Funcionamento

- O atributo **border** permite indicar se uma tabela deve ou não incluir limites (*borders*)
- Por defeito as tabelas não possuem limites

Exemplo

```
<table border="1">  
<tr><th>Nome</th><th>Idade</th></tr>  
<tr><td>João</td><td>25</td></tr>  
<tr><td>Maria</td><td>47</td></tr>  
</table>
```

Nome	Idade
João	25
Maria	47

Células que abrangem várias linhas/colunas

Funcionamento

- Os atributos **rowspan** e **colspan** permitem criar células que abrangem mais do que uma linha e/ou coluna
- Por defeito, cada célula abrange apenas uma linha e uma coluna

Exemplo

```
<table border="1">  
<tr><th colspan="2">Dados</th></tr>  
<tr><th>Nome</th><th>Idade</th></tr>  
<tr><td>João</td><td>25</td></tr>  
<tr><td>Maria</td><td>47</td></tr>  
</table>
```

Dados	
Nome	Idade
João	25
Maria	47

Tabelas com legendas

Funcionamento

- A tag **caption** pode ser utilizada para atribuir uma legenda a uma tabela

Exemplo

```
<table border="1">  
<caption>Legenda</caption>  
<tr><th>Nome</th><th>Idade</th></tr>  
<tr><td>João</td><td>25</td></tr>  
<tr><td>Maria</td><td>47</td></tr>  
</table>
```

Nome	Idade
João	25
Maria	47

1 HTML

- Introdução
- Páginas básicas
- Texto
- Listas
- Imagens
- Links
- Tabelas
- Outras tags

2 CSS

- Introdução
- Selecção de elementos
- Propriedades de formatação

3 Bibliografia

Tag `div`

Funcionamento

- A tag `div` permite criar elementos que dividem o documento HTML em blocos
- Cada bloco pode incorporar vários parágrafos, imagens, tabelas, etc.
- Só por si, não tem nenhum efeito na forma como o browser mostra o documento
- A utilidade de dividir o documento em blocos está em poder atribuir diferentes estilos a cada bloco (como se verá mais adiante)

Exemplo

```
<div>  
  <p>Parte 1</p>  
</div>  
<div><p>Parte 2</p></div>
```

Parte 1

Parte 2

Tag `span`

Funcionamento

- A tag `span` é utilizada para criar um sub-elemento a partir de parte do conteúdo de outro elemento (por exemplo, parte do texto de um parágrafo)
- Só por si, não tem nenhum efeito na forma como o browser mostra o documento
- Mas pode-se aplicar um estilo ao sub-elemento `span` diferente do estilo do resto do elemento

Exemplo

```
<h1>Dia <span>25</span></h1>  
<p>Vai <span>chover</span>.</p>
```

Dia 25

Vai chover.

Tag `hr`

Funcionamento

- A tag `hr` permite criar uma barra horizontal
- Normalmente, estas barras são utilizadas como separadores, e são escritas na forma abreviada (`<hr/>`)

Exemplo

```
<p>Parte 1</p>  
<hr/>  
<p>Parte 2</p>
```

Parte 1



Parte 2

1 HTML

- Introdução
- Páginas básicas
- Texto
- Listas
- Imagens
- Links
- Tabelas
- Outras tags

2 CSS

- Introdução
- Selecção de elementos
- Propriedades de formatação

3 Bibliografia

1 HTML

- Introdução
- Páginas básicas
- Texto
- Listas
- Imagens
- Links
- Tabelas
- Outras tags

2 CSS

- **Introdução**
- Selecção de elementos
- Propriedades de formatação

3 Bibliografia

CSS — *Cascading Style Sheets*

O que são folhas de estilo?

- Uma **folha de estilos**, ou **style sheet**, é formada por um conjunto de regras que indicam ao browser como deve apresentar um documento
- Cada regra é formada por um **selector** e um **estilo**
- O **selector** indica a que elementos é que o estilo deve ser aplicado
- O **estilo** é especificado atribuindo valores a um conjunto de propriedades que se aplicam aos elementos a “estilar”

Exemplo de uma folha de estilos

```
/* Comentário: elementos h1 com letra extra-grande,  
   a vermelho, e elementos h2 com letra grande, a azul */  
  
h1 { font-size: x-large; color: red }  
h2 { font-size: large; color: blue }
```


Aplicação de folhas de estilo a documentos HTML

Formas de aplicar uma folha de estilos

- 1 A folha de estilos é escrita directamente no documento HTML, dentro de um elemento `style`, no cabeçalho do documento
- 2 A folha de estilos é escrita num ficheiro (com extensão `.css`) e é feita uma ligação a esse ficheiro a partir de um elemento `link` no cabeçalho do documento HTML

Exemplo: folha de estilos no documento HTML

Ficheiro HTML

```
...  
<head>  
  ...  
  <style type="text/css">  
    h1 { font-size: x-large; color: red }  
    h2 { font-size: large; color: blue }  
  </style>  
  ...  
</head>  
...
```

Exemplo: folha de estilos num ficheiro `.css`

Ficheiro HTML

```
...  
<head>  
    ...  
    <link  
        media="all"  
        href="styles.css"  
        rel="stylesheet"  
        type="text/css"/>  
    ...  
</head>  
...
```

Ficheiro `styles.css`

```
h1 { font-size: x-large; color: red }  
h2 { font-size: large; color: blue }
```

Herança de estilos

Descrição

- A maioria da formatação aplicada a um elemento é herdada pelos seus descendentes
- Por exemplo, se for atribuído um valor à propriedade **color** no elemento **body**, então esse valor vai ser aplicado nos parágrafos, e no texto dos outros elementos descendentes do elemento **body**
- Existem excepções à regra; por exemplo, as margens definidas no elemento **body** não são herdadas pelos parágrafos

Conflitos e prioridades

Prioridades

- As regras especificadas nas folhas de estilo podem entrar em conflito quanto à formatação a aplicar a alguns elementos
- Por exemplo, a um parágrafo de classe **nota** pode ser aplicada formatação seleccionada por se tratar de um parágrafo (selector: **p**) ou por ser um parágrafo de classe **nota** (selector: **p.nota**)
- Os conflitos são resolvidos de acordo com certas regras, que incluem:
 - ① As regras mais específicas prevalecem sobre as mais gerais; no exemplo anterior, prevaleceria a regra com o selector **p.nota**
 - ② Entre duas regras com a mesma especificidade é dada prioridade à que vem em último lugar

1 HTML

- Introdução
- Páginas básicas
- Texto
- Listas
- Imagens
- Links
- Tabelas
- Outras tags

2 CSS

- Introdução
- **Seleccção de elementos**
- Propriedades de formatação

3 Bibliografia

Selectores

Tipos de selectores

Os selectores utilizados nas regras de formatação podem seleccionar elementos de várias formas, entre elas:

- 1 Pela **tag** dos elementos
- 2 Pelo atributo **class** dos elementos
- 3 Por pseudo-elemento ou pseudo-classe
- 4 Por um contexto formado por vários selectores

Agrupamento de selectores

- Para diminuir as repetições, é possível agrupar selectores e declarações
- Por exemplo, para aplicar uma formatação igual aos elementos **h1**, **h2** e **h3**, escreve-se:

```
h1, h2, h3 { font-family: Arial; color: blue }
```

Seleção por tag

Funcionamento

A seleção de elementos por tag é feita indicando as tags dos elementos a seleccionar

Exemplo: Seleção de todos os parágrafos

```
p { color: blue }
```

Exemplo: Seleção de texto a itálico

```
it { color: red }
```

Exemplo: Seleção dos títulos h1 e h2

```
h1, h2 { font-weight: bold }
```


Seleção pelo atributo `class`

Funcionamento

- O atributo `class` é um atributo especial que pode ser utilizado em qualquer elemento
- A um único elemento podem ser atribuídas várias classes (no mesmo atributo, separadas por espaços)
- Pode-se assim ser aplicar várias regras de formatação, cada uma delas seleccionada com base numa classe

Seleção pelo atributo `class` — Exemplo I

HTML

```
<p class="fundo">Alerta</p>
```

CSS

```
/* fundo amarelo nos parágrafos com classe fundo */  
p.fundo { background-color: yellow }
```



Alerta

Seleção pelo atributo `class` — Exemplo II

HTML

```
<p>Isto é <b>importante</b></p>  
<p>Isto é <span class="vermelho">importante</span></p>  
<p class="vermelho">Isto é importante</p>
```

CSS

```
/* O texto bold aparece a vermelho */  
/* O texto dos elementos com classe vermelho também */  
b, .vermelho { color: red }
```

Isto é **importante**

Isto é **importante**

Isto é **importante**

Seleção pelo atributo `class` — Exemplo III

HTML

```
<p class="fundo vermelho">Este texto  
<span class="nota">aparece</span>  
com fundo amarelo e texto vermelho</p>
```

CSS

```
.nota { font-style: italic; font-size: 16pt }  
.fundo { background-color: yellow }  
.vermelho { color: red }
```

Este texto *aparece* com fundo amarelo
e texto vermelho

Seleção pelo atributo `class` — Exemplo I

HTML

```
<p class="fundo">Alerta</p>
```

CSS

```
/* fundo amarelo nos parágrafos com classe fundo */  
p.fundo { background-color: yellow }
```



Alerta

Seleção por “pseudo-elemento” ou “pseudo-classe”

O que são?

- As **pseudo-classes** e os **pseudo-elementos** são classes e elementos especiais que podem ser utilizados nos selectores
- As **pseudo-classes** permitem distinguir entre diferentes estados dos elementos; são utilizados, por exemplo, para distinguir links já visitados de links ainda não visitados
- Os **pseudo-elementos** referem-se a sub-partes de elementos; por exemplo, à primeira letra de um parágrafo

Como são utilizados?

- O nome da pseudo-classe ou do pseudo elemento é acrescentado ao selector, separado por dois pontos
- Por exemplo, o selector **a:visited** selecciona os links que já foram visitados

Pseudo-classes do elemento `a`

Referência

`link` Refere-se ainda não visitados

`active` Refere-se a links activos (links em que se está a clicar)

`visited` Refere-se a links já visitados

`hover` Refere-se a links que têm o ponteiro do rato por cima

Exemplos

```
/* Pelas regras da prioridade, um link activo que já  
   tenha sido visitado aparece a azul, e não a verde */
```

```
a:link      { color: red }  
a:visited   { color: green; font-size: 85% }  
a:active    { color: blue; font-size: 125% }  
a:hover     { color: #000; font-size: 125% }
```

Pseudo-elementos `first-letter` e `first-line`

Referência

`first-line` Refere-se à primeira linha de texto do elemento

`first-letter` Refere-se à primeira letra de texto do elemento

Exemplos

```
/* A primeira linha aparece a negrito e  
   a primeira letra aparece maior */
```

```
p:first-line { font-weight: bold }
```

```
p:first-letter { font-size: 300% }
```


Seleccção por contexto

Funcionamento

- A selecção de elementos por contexto é feita indicando vários selectores separados por espaços
- Esses selectores indicam uma hierarquia
- Para um elemento ser seleccionado, deve pertencer a uma hierarquia “compatível” com a indicada no selector

Exemplo I

```
/* elementos it pertencentes a parágrafos */  
p it { color: red }
```

Exemplo II

```
/* elementos com classe nota pertencentes a títulos h1 */  
h1 .nota { color: blue }
```

1 HTML

- Introdução
- Páginas básicas
- Texto
- Listas
- Imagens
- Links
- Tabelas
- Outras tags

2 CSS

- Introdução
- Selecção de elementos
- **Propriedades de formatação**

3 Bibliografia

Tamanhos e unidades de medida

Tamanhos

Um valor de tamanho é formado por um `-` ou `+` opcional, seguido por um número e duas letras que indicam a unidade de medida utilizada (sem espaços entre o número e a unidade).

Unidades relativas

- `1em` é a altura normal dos caracteres
- `1ex` é a altura da letra `x`
- `1px` é um pixel

Unidades absolutas

- `1in` é uma polegada, e $1in=2.54cm$
- `1cm` é um centímetro
- `1mm` é milímetro
- `1pt` é um ponto, e $1pt=1/72in$
- `1pc` é um pica, e $1pc=12pt$

Propriedade `font-family`

Funcionamento

- Indica a fonte a ser utilizada (tipo de letra)
- Pode-se indicar mais do que uma fonte; o browser começa pela esquerda até encontrar uma que possa utilizar

Valores possíveis

- O nome de uma fonte (por exemplo, `Arial`, ou `"Times New Roman"`)
- Uma categoria de fontes (por exemplo, `serif`, `sans-serif`, ou `monospace`)

Exemplos

```
h1 { font-family: sans-serif }
```

```
p { font-family: "New Century Schoolbook", Times, serif }
```

Propriedade `font-style`

Funcionamento

- Indica se o texto deve ser **itálico**, **oblíquo**, ou **normal**

Valores possíveis

- **normal**, **italic** ou **oblique**
- O valor por defeito é **normal**

Exemplos

```
h3 { font-style: italic }
```

Propriedade `font-weight`

Funcionamento

- Indica se o texto deve ser mais claro ou mais escuro (negrito)

Alguns valores possíveis

- `normal`, `bold`, `bolder` ou `lighter`

Exemplos

```
h1 { font-weight: bold }  
p  { font-weight: normal }
```

Propriedade `font-size`

Funcionamento

- Indica o tamanho do texto (dos caracteres)

Alguns valores possíveis

- Um dos valores `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large` ou `xx-large`
- Um tamanho ou uma percentagem
- A percentagem é em relação ao tamanho do texto no elemento-pai

Exemplos

```
h1 { font-size: xx-large }  
p  { font-size: 12pt }
```

Propriedade `color`

Funcionamento

- Indica a cor dos elementos (por exemplo, do texto)

Alguns valores possíveis

- Uma das seguintes cores, identificadas por nome: `aqua`, `black`, `blue`, `fuchsia`, `gray`, `green`, `lime`, `maroon`, `navy`, `olive`, `purple`, `red`, `silver`, `teal`, `white` ou `yellow`
- Uma cor em formato `rgb`: `#rrggbb` ou `#rgb`

Exemplos

```
h1 { color: olive }  
p.nota { color: #fff800 }  
b { color: #000 }
```


Propriedade `background-color`

Funcionamento

- Indica a cor de fundo dos elementos

Valores possíveis

- O valor `transparent`, mais os valores possíveis para a propriedade `color`
- O valor por defeito é `transparent`

Exemplos

```
h1 { background-color: transparent }  
p.nota { background-color: #ccc }
```

Propriedade `background-image`

Funcionamento

- Indica a imagem de fundo dos elementos

Valores possíveis

- O `URL` de uma imagem

Exemplos

```
body { background-image: url(/images/foo.gif) }  
p { background-image: url(http://www.htmlhelp.com/bg.png) }
```

Propriedade `background-repeat`

Funcionamento

- Indica como uma imagem de fundo é repetida

Valores possíveis

- Um dos seguintes valores: `repeat`, `repeat-x`, `repeat-y` ou `no-repeat`
- O valor por defeito é `repeat`

Exemplos

```
body { background-repeat: repeat-x }  
table { background-repeat: repeat }
```

Propriedade `text-decoration`

Funcionamento

- Indica como é que o texto deve ser “decorado”

Valores possíveis

- O valor `none`, ou uma combinação dos seguintes valores: `underline`, `overline`, `line-through` e `blink`
- O valor por defeito é `none`

Exemplos

```
.nota { text-decoration: blink }  
.apagado { text-decoration: line-through }
```

Propriedade `vertical-align`

Funcionamento

- Indica o alinhamento vertical de um sub-elemento em relação

Alguns valores possíveis

- Um dos seguintes valores: `baseline`, `sub`, `super`, `top`, `text-top`, `middle`, `bottom` ou `text-bottom`
- Um valor numérico indicado em percentagem
- O valor por defeito é `baseline`

Exemplos

```
img.meio { vertical-align: middle }  
img { vertical-align: 50% }  
.expoente { vertical-align: super }
```

Propriedade `text-align`

Funcionamento

- Indica como é que o texto deve ser alinhado (na horizontal)

Alguns valores possíveis

- Um dos seguintes valores: `left`, `right`, `center` ou `justify`

Exemplos

```
h1      { text-align: center }  
p.nota { text-align: justify }
```

Propriedade `text-indent`

Funcionamento

- Indica o avanço da primeira linha de texto do elemento

Valores possíveis

- Uma percentagem ou um tamanho
- A percentagem é em relação à largura do elemento-pai
- O valor por defeito é `0pt`

Exemplos

```
/* avançar a primeira linha dos parágrafos */  
p { text-indent: 5em }
```

Propriedade `text-transform`

Funcionamento

- Indica se o texto do elemento deve ser convertido para maiúsculas (todo ou só a primeira letra de cada palavra) ou para minúsculas

Valores possíveis

- Um dos seguintes valores: `none`, `capitalize`, `uppercase` ou `lowercase`
- O valor por defeito é `none`

Exemplos

```
h1 { text-transform: uppercase }  
h2 { text-transform: capitalize }
```


Propriedade `line-height`

Funcionamento

- Indica a altura das linhas de texto

Valores possíveis

- O valor `normal`, um valor numérico sem unidades, um tamanho (valor numérico com unidades) ou uma percentagem
- O valor por defeito é `normal`
- Se o valor numérico não tiver unidades, então a altura da linha será a sua altura normal, multiplicada pelo valor indicado
- A percentagem é em relação à altura normal da linha

Exemplos

```
/* texto com espaçamento de duas linhas */  
p { line-height: 200% }
```

Propriedades `margin-top`, `margin-bottom`, `margin-left`, `margin-right` e `margin`

Funcionamento

- Tamanho de uma ou todas as margens do elemento (com `margin`)
- Na propriedade `margin` a ordem é `top-right-bottom-left`; indicando-se um só valor, as margens são todas iguais; indicando-se mais, as margens em falta são iguais às dos lados opostos

Valores possíveis

- Um tamanho (`0` por defeito), uma percentagem (em relação à largura do elemento-pai), ou o valor `auto`

Exemplos

```
body { margin: 5em } /* todas iguais */  
p { margin: 2em 4em } /* top/bottom 2em, left/right 4em */
```

Propriedades `padding-top`, `padding-bottom`, `padding-left`, `padding-right` e `padding`

Funcionamento

- Tamanho do espaço entre a margem do elemento e o seu conteúdo
- Na propriedade `padding` aplicam-se as mesmas regras da propriedade `margin`

Valores possíveis

- Um tamanho (`0` por defeito), ou uma percentagem (em relação à largura do elemento-pai)

Exemplos

```
body { padding: 10px } /* igual nos 4 lados */  
span.nota { padding-top: 2em; padding-bottom: 1em }
```

Propriedades `border-top-width`,
`border-bottom-width`, `border-left-width`,
`border-right-width` e `border-width`

Funcionamento

- Tamanho dos limites (*borders*) do elemento
- Na propriedade `border-width` aplicam-se as mesmas regras da propriedade `margin`

Valores possíveis

- Um dos seguintes valores: `thin`, `medium` ou `thick`
- Um tamanho

Exemplos

```
h1 { border-width: 2px }  
p  { border-width: 2px 4px }
```

Propriedade `border-color`

Funcionamento

- Indica as cores utilizadas nos limites (*borders*) do elemento
- Podem-se indicar entre uma e quatro cores, aplicando-se as mesmas regras da propriedade `margin`

Valores possíveis

- Uma a quatro cores, identificadas por nome ou por `rgb`

Exemplos

```
h1 { border-color: red }  
p  { border-color: #ccc }
```

Propriedade `border-style`

Funcionamento

- Indica o estilo dos limites (*borders*) do elemento (sólido, tracejado, pontos, etc.)
- Podem-se indicar entre um e quatro valores, aplicando-se as mesmas regras da propriedade `margin`

Valores possíveis

- Um a quatro valores de entre os valores possíveis: `none`, `dotted`, `dashed`, `solid`, `double`, `groove`, `ridge`, `inset` e `outset`

Exemplos

```
h1 { border-style: dotted }  
p  { border-style: solid }
```

Propriedades `border-top`, `border-bottom`, `border-left` e `border-right`

Funcionamento

- Permitem indicar ao mesmo tempo a largura, estilo e cor de um dos limites do elemento
- A ordem é `width-style-color`

Exemplos

```
h1 { border-bottom: 1px solid black }  
p  { border-left:  1mm dotted red   }
```

Propriedade `border`

Funcionamento

- Permitem indicar ao mesmo tempo a largura, estilo e cor de todos os limites do elemento
- Os limites serão iguais nos quatro lados
- A ordem é `width-style-color`

Exemplos

```
h1 { border: 1px solid black }  
p  { border: 1mm dotted red }
```


Propriedades `width` e `height`

Funcionamento

- Indicam a largura e a altura do elemento (não é aplicável a todos os elementos)

Valores possíveis

- Um tamanho, uma percentagem (só para `width`, e em relação à largura do elemento-pai), ou o valor `auto` (o valor por defeito)

Exemplos

```
.nota { width: 10em }
```

Propriedades `float` e `clear`

Funcionamento

- A propriedade `float` indica que o elemento é flutuante, isto é, que pode ser rodeado pelo texto de outros elementos
- A propriedade `clear` indica em que lados é que o elemento não pode ser rodeado por elementos flutuantes

Valores possíveis

- Propriedade `float`: um dos valores `left`, `right` ou `none` (valor por defeito)
- Propriedade `clear`: um dos valores `left`, `right`, `both` ou `none` (valor por defeito)

Exemplos

```
.nota { float: left; width: 5cm; clear: left }
```

Propriedade `display`

Funcionamento

Indica como deve ser visualizado o elemento:

- `block` Como um bloco, com quebras de linha antes e depois
- `inline` Sem quebras de linha
- `item-list` Como elemento de uma lista, com quebras de linha antes e depois
- `none` O elemento e os seus descendentes não são mostrados

Exemplos

```
.esconder { display: none }
```

1 HTML

- Introdução
- Páginas básicas
- Texto
- Listas
- Imagens
- Links
- Tabelas
- Outras tags

2 CSS


- Introdução
- Selecção de elementos
- Propriedades de formatação

3 Bibliografia

Bibliografia

 **HTML & XHTML: the definitive guide**
Chuck Musciano & Bill Kennedy, O'Reilly, 2000
[Cota 2408 na biblioteca do DCC](#)

 **HTML Tutorial**
<http://www.w3schools.com/html/default.asp>

 **Página sobre HTML na Wikipedia**
<http://en.wikipedia.org/wiki/HTML>

 **Página sobre XHTML na Wikipedia**
<http://en.wikipedia.org/wiki/XHTML>

 **Guide to Cascading Style Sheets**
<http://www.htmlhelp.com/reference/css/>