

# CSS

## Cascading Style Sheets

## CSS

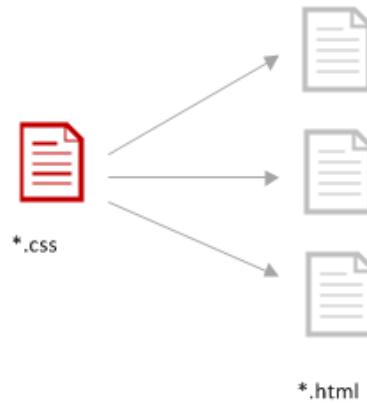
- Incremento da qualidade e consistência gráfica das aplicações
- Separação do **design e do conteúdo**:
  - **HTML** define a estrutura (conteúdo)
  - **CSS** define a formatação/design
    - Layouts
    - Posicionamentos
    - Cores
    - Texto
    - Formulários
    - Tabelas
    - Listas
    - ...

\*



## ■ Vantagens

- Possibilidade de aplicação imediata a vários documentos HTML
  - Garante que diversos documentos HTML são submetidos às mesmas regras de formatação.
- Muito maior flexibilidade
  - Alterações centralizadas
  - Evita a repetição de formatações
- Maior correção
  - Melhora a portabilidade
  - Reduz a possibilidade de erros
  - Maior uniformidade
    - Melhora a consistência gráfica



## Sintaxe CSS

- Composta por:
  - Seletor
  - Declaração /conjunto de declarações
    - Propriedade
    - Valor

*seletor { propriedade:valor;*

- Cada regra seleciona um elemento/conjunto de elementos e declara toda a formatação associada
- A sintaxe das CSS é **case sensitive**

### ■ Seletor

- Elemento crucial na definição de uma regra CSS
  - A compreensão do funcionamento dos seletores é absolutamente essencial para permitir uma correta implementação de CSS
- Permite identificar os elementos HTML a que a regra CSS se aplica



```
body{ background-color:#FFF;}
```

### ■ Declaração

- Uma atribuição de um valor a uma propriedade. Uma regra pode conter várias declarações (*declaration block*).
  - Propriedade
    - Atributo da folha de estilo ao qual deve ser atribuído um valor (ex: *color*, ...)

```
body{ background-color:#FFF;}
```

- Valor

- Especificação concreta da variável (ex: *arial*, *#0000FF*)

```
body{ background-color:#FFF;
```

Sintaxe	Exemplo
selector { propriedade: valor}	body {background-color: #FFFFFF}
<i>bloco de declarações ;</i>	
selector { propriedade: valor; propriedade: valor; propriedade: valor; }	p { text-align:center; color:red; font-family: arial;} 
<i>seleção múltipla ,</i>	
selector1, selector2, selector3 { propriedade: valor;}	h1, h2, h3{ color:red;}

## Ligação CSS / HTML

## Ligação CSS / HTML

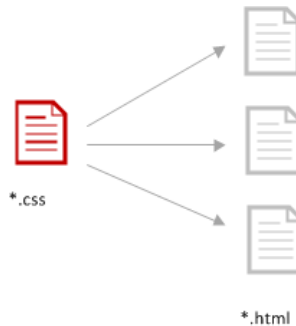
### ■ Como aplicar estilos (regras CSS) aos documentos HTML?

#### ■ **External Style Sheet**

- Ficheiro externo de extensão \*.css que contém as regras CSS
- Pode ser aplicado a diversos documentos HTML
- Forma mais poderosa e flexível de incorporar CSS

styles.css

```
body{  
  height:100%;  
  font-size:1rem;  
  font-family: 'Montserrat', sans-serif;  
}
```



index.html

```
<link rel="stylesheet" href="css/styles.css" >
```

## Ligação CSS / HTML

### ■ **External Style Sheet (\*.css)**

- Um ficheiro CSS (\*.css) é exclusivamente constituído por regras CSS
- O código CSS não contém tags

seletor{propriedade:valor;}

```
/* style1.css */
```

```
p { font-weight:bold;  
    color:#F00;}
```

### ■ `<link ... />`

- Atributo **href** indica a localização do ficheiro \*.css . Atributo obrigatório!
- Atributo **rel** define a relação do documento HTML com o ficheiro externo, neste caso é *stylesheet*. Atributo obrigatório!
- Atributo **type** é opcional na última versão do HTML5 (legado de versões anteriores)

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8" />
<title> External Style Sheet</title>

<link href="style1.css" rel="stylesheet" type="text/css"/>

</head>

<body>
</body>
</html>
```

```
/* style1.css */

p { font-weight:bold;
    color:#F00;}
```

### ■ **External Style Sheet**

- através de **@import** e da tag **<style>**

```
<!DOCTYPE HTML>
<html>
<head>
  <style>
    @import url("style1.css");
    ...
  </style>
  <meta charset="utf-8" />
  <title> External Style Sheet</title>
</head>

<body>
</body>
</html>
```

```
/* style1.css */

p { font-weight:bold;
    color:#F00;}
```

- as duas formas de estabelecer a ligação a um ficheiro externo são suportadas pela maioria dos browsers originando resultados semelhantes, no entanto:
  - A tag **<link>** é a forma mais eficaz, aquela que assegura melhor performance, para efetuar a ligação de uma HTML a um CSS externo

### ■ **Embedded Style Sheet**

- Necessita de ser definido em todos os documentos \*.html
  - Menos flexível e eficaz do que a ligação a um ficheiro externo
  - Aumenta o peso dos ficheiros
  - Uma alteração na formatação implica uma alteração em cada um dos ficheiros
    - em MPA pode originar inconsistência na formatação entre diferentes \*.html



### ■ **<style> ... </style>**

- definido do <head> do \*.html
  - Atributo type="text/css" é obrigatório nas versões anteriores ao HTML5.

Como é óbvio **não podem** ser definidas tags HTML nesta janela, trata-se de código CSS

```
<!DOCTYPE HTML>
<html>
<head>
  <style>
    p { font-weight:bold;
        color:#F00;}
  </style>
  <meta charset="utf-8" />
  <title> Embedded Style Sheet</title>
</head>

<body>
</body>
</html>
```

Janela de  
código CSS

### ■ *Inline Style*

- aplicar o estilo localmente através do atributo **style**
  - É a forma **menos flexível** e **mais propensa** a erros para formatar um elemento HTML
  - Se efetuado de forma estática, favorece a inconsistência na formatação dos conteúdos
  - Implica tantas alterações quantas as definições efetuadas

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
</head>
<body>
  <h1 style="color:orange">Inline Style: A evitar!</h1>
</body>
</html>
```

**Inline Style: A evitar!**

## Seletores CSS



## Seletor CSS

- Permite identificar os elementos HTML aos quais a regra CSS se aplica

*seletor { propriedade:valor;}*

- **Diferentes tipos de seletores** de forma a permitir a flexibilidade necessária para selecionar qualquer dos elementos/conjunto de elementos definidos no HTML
- Os seletores CSS são cruciais para a aplicação da tecnologia CSS, mas também são usados pelo JavaScript / Frameworks JS (ex: React) para aceder a determinado elemento HTML.

Seletores CSS **não se esgotam** na tecnologia CSS!

- Devem ser definidos tendo sempre em consideração os conflitos de formatação:
  - Muito frequentes sempre que o código CSS ganha alguma escala.

## Seletor CSS

- Principais tipos de seletores:

*Elemento*

*Contexto*

*id*

*class*

*Atributo*

- Principais tipos de seletores:

*Elemento*

## Seletor CSS

- Seletor de **Elemento**

*Elemento*

- Referência direta ao elemento HTML que se pretende formatar
  - Vantagem: simplicidade
  - Desvantagem: pouco específicos

```
<style>  
p{border:1px solid blue;}  
h2{color:blue}  
</style>
```

- Elementos Agrupados

- Referência direta aos elementos HTML que se pretendem formatar, os quais devem ser separados por **vírgula (,)**

```
<style>  
h1,h2{ color: #900;}  
</style>
```

## Seletor CSS

- Seletor Universal (\*)
  - Permite selecionar todos os elementos
  - A utilização deste seletor deve ser cuidadosamente ponderada é aplicado a **todos os elementos**
    - podem ser originadas algumas formatações/efeitos não previstos

```
<style>  
  * {color:orange}  
</style>  
</head>  
  
<body>  
  <p>p first-child</p>  
  <ul>  
    <li>first option</li>  
    <li>second option</li>  
  </ul>  
  <p>p last-child</p>  
</body>
```

p first-child

- first option
- second option

p last-child

## Seletor CSS

- Principais tipos de seletores:

*Contexto*

## Seletor CSS

### ■ Seletor de **Contexto**

#### ■ Descendentes

- A formatação dos elementos depende do contexto
- Elementos separados por **espaço em branco**

Contexto

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>

  <style>
    li em {color:orange;}
  </style>
</head>
<body>
  <ul>
    <li><em>1ª opção</em></li>
    <li><em>2ª opção</em></li>
    <li><em>3ª opção</em></li>
  </ul>
  <p>1º <em>parágrafo</em></p>
  <p>2º <em>parágrafo</em></p>
</body>
</html>
```

- 1ª opção
- 2ª opção
- 3ª opção

1º parágrafo

2º parágrafo

Podem ser criados vários  
níveis de dependências

```
<style>
  ul li em {color:orange;}
</style>
```

## Seletor CSS

### ■ Seletores de **Contexto**

#### ■ Casos particulares de seletores de contexto

- Descendentes diretos (**filhos**) - *child selector* (>)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>

  <style>
    span {color:gray;}
    p > span {color:orange;font-size:1.2em}
  </style>
</head>
<body>
  <p><span>1º <span>parágrafo</span></span></p>
  <p><span>2º <span>parágrafo</span></span></p>
</body>
</html>
```

1º parágrafo

2º parágrafo

## Seletor CSS

### ■ Seletores de **Contexto**

#### ■ Elementos Adjacentes (**Adjacent Sibling Selector**) (+)

- Adjacente: elemento imediatamente seguinte ao elemento que define o contexto

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    → h1 + p {color:orange;font-size:1.2em}
  </style>
</head>
<body>
  <h1 id="p1"> Referência</h1>
  <p><span>1º <span>parágrafo</span></span></p>
  <p><span>2º <span>parágrafo</span></span></p>
</body>
</html>
```

## Referência

1º parágrafo

2º parágrafo

## Seletor CSS

### ■ Seletores de **Contexto**

#### ■ *element1 ~ element2*

- seleciona todos os *element2* que são precedidos por um *element1*
  - Os elementos devem ter o mesmo pai
  - O *element2* não tem de ser imediatamente precedido pelo *element1*

```
<!DOCTYPE html>
<html lang="">
<head>
  <meta charset="utf-8">
  <title></title>

  <style>
    → h1 ~ p {color:orange}
  </style>
</head>
<body>
  <p> 1º parágrafo </p>
  <h1 id="heading1"> Referência </h1>
  <p> 2º parágrafo </p>
  <p> 3º parágrafo </p>
</body>
</html>
```

1º parágrafo

## Referência

2º parágrafo

3º parágrafo

- Principais tipos de seletores:

*id*

## Seletor CSS

- Selectores de ID (#)

- Atribuir um estilo a uma única ocorrência de um elemento HTML

*id*

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    p {color:gray;}
    #p1 {color:orange;font-size:1.2em}
  </style>
</head>
<body>
  <p id="p1"> Parágrafo com id=p1</p>
  <p><span>1º</span> <span>parágrafo</span></span></p>
  <p><span>2º</span> <span>parágrafo</span></span></p>
</body>
</html>
```

Parágrafo com id=p1

1º parágrafo

2º parágrafo

- Os id's são únicos e como tal dispensam a especificação do elemento
  - seletores equivalentes
    - Especificidade diferente

#p1    p#p1

## Seletor CSS

- Seletor de ID (#)
  - Podem ser utilizados como parte de um seletor de contexto

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    li {color:gray;}
    #lista1 li {color:orange;}
  </style>
</head>
<body>
  <ul id="lista1">
    <li>1ª opção / lista id="lista1" </li>
    <li>2ª opção / lista id="lista1" </li>
  </ul>
  <ul>
    <li>1ª opção / lista sem id </li>
    <li>2ª opção / lista sem id </li>
  </ul>
</body>
</html>
```

- 1ª opção / lista id="lista1"
- 2ª opção / lista id="lista1"
- 1ª opção / lista sem id
- 2ª opção / lista sem id

## Seletor CSS

- Principais tipos de seletores:

*class*

## Seletor CSS

### ■ Seletor de *class* .

- Ao contrário do atributo ID:
  - o atributo *class* pode ser partilhada por múltiplos elementos
- Um elemento pode ter definida mais de uma *class*
  - Sem limite, separadas por espaço, sendo indiferente a ordem pela qual são definidas.

*class*

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    li {color:gray;}
    .opc {color:orange;}
  </style>
</head>
<body>
  <ul id="lista1">
    <li class="opc">1ª opção / lista id="lista1" </li>
    <li class="opc">2ª opção / lista id="lista1" </li>
  </ul>
  <ul>
    <li class="opc">1ª opção / lista sem id </li>
    <li>2ª opção / lista sem id </li>
  </ul>
</body>
</html>
```

- 1ª opção / lista id="lista1"
- 2ª opção / lista id="lista1"
- 1ª opção / lista sem id
- 2ª opção / lista sem id

## Seletor CSS

### ■ Seletor de *class* .

- *class* aplicada/definida com base num elemento específico

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    li {color:gray;}
    p.opc {color:orange;}
  </style>
</head>
<body>
  <ul id="lista1">
    <li class="opc">1ª opção / lista id="lista1" </li>
    <li class="opc">2ª opção / lista id="lista1" </li>
  </ul>
  <ul>
    <li class="opc">1ª opção / lista sem id </li>
    <li>2ª opção / lista sem id </li>
  </ul>
  <p class="opc">1º Parágrafo</p>
</body>
</html>
```

- 1ª opção / lista id="lista1"
- 2ª opção / lista id="lista1"
- 1ª opção / lista sem id
- 2ª opção / lista sem id

1º Parágrafo



## Seletor CSS

- Seletor de *class*.
  - Podem ser utilizados para criar um seletor contextual
    - São formatados os elementos integrados num outro elemento cujo atributo *class* foi definido com um valor específico.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    li {color:gray;}
    .lst .opc {color:orange;}
  </style>
</head>
<body>
  <ul id="lista1">
    <li class="opc">1ª opção / lista id="lista1" </li>
    <li class="opc">2ª opção / lista id="lista1" </li>
  </ul>
  <ol class="lst">
    <li class="opc">1ª opção / lista sem id </li>
    <li>2ª opção / lista sem id </li>
  </ol>
  <p class="opc">1º Parágrafo</p>
</body>
</html>
```

- 1ª opção / lista id="lista1"
- 2ª opção / lista id="lista1"

1. 1ª opção / lista sem id
2. 2ª opção / lista sem id

1º Parágrafo

## Seletor CSS

- **Importante!**
  - Especificação do elemento e seletor **id**

**h1#d1{color:gray}**

Formatação aplicada ao elemento **h1** cujo **id** é **d1**

≠

h1 #d1

- Especificação do elemento e seletor **class**

**h2.c1{color:green}**

Formatação aplicada ao elemento **h2** cuja **class** é **c1**

≠

h2 .c1

O espaço em branco permite implementar seletores totalmente diferentes

## Seletor CSS

```
...
<style>
  h2.c1{color:orange}
</style>

</head>

<body>
  <div id="div1" class="c1">
    <h1 id="d1">heading 1</h1>
    <h2 class="c1">heading 2</h2>
    <p class="c1">paragraph1</p>
  </div>
  <h2 class="c1">Other heading 2</h2>
...

```

Formatação aplicada aos elementos h2  
cuja **class** é **c1**

heading 1  
heading 2  
paragraph1  
Other heading 2

≠

A ordem dos elementos / espaço  
em branco é determinante para  
uma correta implementação de um  
selector.

```
...
<style>
  .c1 h2{color:orange}
</style>

</head>

<body>
  <div id="div1" class="c1">
    <h1 id="d1">heading 1</h1>
    <h2 class="c1">heading 2</h2>
    <p class="c1">paragraph1</p>
  </div>
  <h2 class="c1">Other heading 2</h2>
...

```

Formatação aplicada  
aos elementos h2  
definidos **no**  
**contexto** de um  
outro elemento cuja  
**class** é **c1**

heading 1  
heading 2  
paragraph1  
Other heading 2

## Seletor CSS

- Principais tipos de seletores:

Atributo

## Seletor CSS

### ■ Seletor de Atributo

#### ■ [attribute]

Atributo

- Caso o elemento não seja especificado são afetados todos os elementos que possuem o atributo definido

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    [id] {color:orange;}
  </style>
</head>
<body>
  <p id="primeiro">1º Parágrafo</p>
  <p> 2º Parágrafo</p>
  <p id="terceiro">3º Parágrafo</p>
  <p> 4º Parágrafo</p>

  <ul id="lista"><li>primeira opção</li></ul>
</body>
</html>
```

1º Parágrafo  
2º Parágrafo  
3º Parágrafo  
4º Parágrafo

- primeira opção

## Seletor CSS

### ■ Attribute Selectors

- Referenciam um elemento através dos seus atributos ou dos valores que assumem

#### ■ **element [attribute]**

- referencia os elementos especificados que têm declarado um atributo específico.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    p[id] {color:orange;}
  </style>
</head>
<body>
  <p id="primeiro">1º Parágrafo</p>
  <p> 2º Parágrafo</p>
  <p id="terceiro">3º Parágrafo</p>
  <p> 4º Parágrafo</p>

  <ul id="lista"><li>primeira opção</li></ul>
</body>
</html>
```

1º Parágrafo  
2º Parágrafo  
3º Parágrafo  
4º Parágrafo

- primeira opção

## Seletor CSS

### ■ ***element* [attribute="value"] / [atribute="value"]**

- referencia os elementos *element* cujo atributo assume um valor *específico*

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    [id="lista"]{color:orange;}
  </style>
</head>
<body>
  <p id="primeiro">1º Parágrafo</p>
  <p> 2º Parágrafo</p>
  <p id="terceiro">3º Parágrafo</p>
  <p> 4º Parágrafo</p>

  <ul id="lista"><li>primeira opção</li></ul>
</body>
</html>
```

1º Parágrafo

2º Parágrafo

3º Parágrafo

4º Parágrafo

- primeira opção

- Existem algumas variantes de seletor de atributo (menos utilizados) que selecionam os elementos a formatar com base em correspondências parciais dos valores dos atributos.

## Seletor CSS

### ■ ***Attribute Selectors***

#### ■ ***element* [attribute \*= " ..."] / [atribute \*= " ..."]**

- referencia os elementos cujos valores do atributo especificado contêm uma dada **expressão** (palavra isolada ou inserida em outra palavra)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    p[id*="paragrafo"] {color:orange;}
  </style>
</head>
<body>
  <p id="paragrafo-primeiro">1º Parágrafo</p>
  <p id="segundo paragrafo">2º Parágrafo</p>
  <p id="paragrafo terceiro">3º Parágrafo</p>
  <p id="quarto">4º Parágrafo</p>
</body>
</html>
```

1º Parágrafo

2º Parágrafo

3º Parágrafo

4º Parágrafo

## Seletor CSS

### ■ Attribute Selectors

#### ■ **element [attribute ^= "..."] / [attribute ^= "..."]**

- referencia os elementos cujo valor do atributo **se inicia** com uma dada **expressão** (palavra isolada ou inserida em outra palavra)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    → p[id^="paragrafo"] {color:orange;}
  </style>
</head>
<body>
  <p id="paragrafo-primeiro">1º Parágrafo</p>
  <p id="segundo paragrafo">2º Parágrafo</p>
  <p id="paragrafo terceiro">3º Parágrafo</p>
</body>
</html>
```

1º Parágrafo

2º Parágrafo

3º Parágrafo

## Seletor CSS

### ■ Attribute Selectors

#### ■ **element [attribute \$= "..."] / [attribute \$= "..."]**

- referencia os elementos cujo valor do atributo **termina com** uma dada **expressão** (palavra isolada ou inserida em outra palavra)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    → p[id$="paragrafo"] {color:orange;}
  </style>
</head>
<body>
  <p id="paragrafo-primeiro">1º Parágrafo</p>
  <p id="segundo paragrafo">2º Parágrafo</p>
  <p id="paragrafo terceiro">3º Parágrafo</p>
</body>
</html>
```

1º Parágrafo

2º Parágrafo

3º Parágrafo

- Lista completa: [http://www.w3schools.com/cssref/css\\_selectors.asp](http://www.w3schools.com/cssref/css_selectors.asp)

```
<body>  
  <p class="cl1"> Parágrafo Exemplo <span> Elemento Inline </span></p>  
</body>
```

<style>

p {color:red;}

Parágrafo Exemplo Elemento Inline

span {color:blue;}

Parágrafo Exemplo Elemento Inline

p span {color:green;}

Parágrafo Exemplo Elemento Inline

p span {color:green;}

Parágrafo Exemplo Elemento Inline

p .cl1 {color:orange;}

Parágrafo Exemplo Elemento Inline

p.cl1 {color:orange;}

Parágrafo Exemplo Elemento Inline

</style>

## Pseudo-Class Selectors

## Pseudo-Class Selector

### ■ Baseados no **estado** do elemento

- *:link*
  - link não visitado
- *:visited*
  - link anteriormente visitado
- *:hover*
  - o cursor do rato encontra-se sobre o elemento
- *:active*
  - o elemento está a ser selecionado
- *:focus*
  - elemento selecionado e pronto para receber valores
- *element:first-child*
  - elemento que é o primeiro filho do seu pai (**excepção!**)

## Pseudo-Class Selector

### ■ *Pseudo-Class selector (exemplo)*

#### ■ *:hover*

- Sobreposição do cursor do rato

Pseudo-Class: estado link

Pseudo-Class: **estado link**

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    a:hover {color:white;background-color:#FF6600;font-size:20px;
  </style>
</head>
<body>
  <p>Pseudo-Class: <a href="http://www.isec.pt">estado link</a></p>
</body>
</html>
```

## Pseudo-Class Selector

### ■ Pseudo-Class selector (exemplo)

#### ■ *li:first-child*

- seleciona o elemento que é o primeiro filho do seu pai

```
<!DOCTYPE html>

<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    li:first-child {color:orange;font-size:20px}
  </style>
</head>
<body>
  <ul>
    <li>Primeiro filho de ul</li>
    <li>Segundo filho de ul</li>
    <li>Último filho de ul</li>
  </ul>
  <p> primeiro parágrafo </p>
</body>
</html>
```

- Primeiro filho de ul
- Segundo filho de ul
- Último filho de ul

primeiro parágrafo

Seletor *sempre aplicado* no elemento filho

## Pseudo-Class Selector

#### ■ *li:last-child*

- seleciona o elemento que é o último filho do seu pai

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    li:last-child {color:orange;font-size:20px}
  </style>
</head>
<body>
  <ul>
    <li>Primeiro filho de ul</li>
    <li>Segundo filho de ul</li>
    <li>Último filho de ul</li>
  </ul>
  <p> primeiro parágrafo </p>
</body>
</html>
```

- Primeiro filho de ul
- Segundo filho de ul
- Último filho de ul

primeiro parágrafo



## Pseudo-Class Selector

### ▪ `:first-child` ; `:last-child`

```
<!DOCTYPE html>
<html lang="">
<head>
  <meta charset="utf-8">
  <title></title>

  <style>
    div:first-child{color:orange}
    div:last-child{color:lightblue}
  </style>
</head>
<body>
  <div> Primeiro elemento div</div>
  <div> Ultimo elemento div</div>
</body>
</html>
```

Primeiro elemento div

Ultimo elemento div

Alteração da  
estrutura  
mantendo o  
CSS

```
<body>
  <p>primeiro filho</p>
  <div> Primeiro elemento div</div>
  <div> Ultimo elemento div</div>
  <p>ultimo filho</p>
</body>
```

?

primeiro filho

Primeiro elemento div  
Ultimo elemento div

ultimo filho

A formatação não é aplicada, uma vez que a alteração da estrutura originou **que os elementos `<div>` deixassem de ser o primeiro e o ultimo filho do seu pai** (neste caso o element `<body>`)

## Seletor CSS

### ▪ `li:nth-child(n)`

- seleciona o elemento que é o enésimo filho

```
<!DOCTYPE html>

<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    li:nth-child(2) {color:orange;font-size:20px}
  </style>
</head>
<body>
  <ul>
    <li>Primeiro filho de ul</li>
    <li>Segundo filho de ul</li>
    <li>Último filho de ul</li>
  </ul>
  <p> primeiro parágrafo </p>
</body>
</html>
```

- Primeiro filho de ul
- **Segundo filho de ul**
- Último filho de ul

primeiro parágrafo

## Pseudo-Class Selector

- `:first-of-type` ; `:last-of-type`; `:nth-of-type(n)`

```
<style>
  div:first-of-type{color:orange}
  div:last-of-type{color:lightblue}
</style>

</head>
<body>
  <p>primeiro filho</p>
  <div>primeiro elemento</div>
  <div>ultimo elemento</div>
  <p>ultimo filho</p>
```

primeiro filho

primeiro elemento

ultimo elemento

ultimo filho

- Apesar de na maioria das situações produzirem resultados idênticos, os seletores:

- `:first-of-type` e `:first-child`
- `:last-of-type` e `:last-child`
- `:nth-child(n)` e `:nth-of-type(n)`

**não são** equivalentes!

## Seletor CSS

- Formulários

- *Pseudo Class Selectors* **específicos** para formulários

<i>seletor</i>	<i>Exemplo</i>	<i>Observações</i>
<code>:checked</code>	<code>input:checked</code>	seleciona todos os input checked
<code>:disabled</code>	<code>input:disabled</code>	seleciona todos os input que estão inativos
<code>:enabled</code>	<code>input:enable</code>	seleciona todos os input que estão ativos
<code>:focus</code>	<code>input:focus</code>	seleciona todos os input que “ganham” <i>focus</i>
<code>:invalid</code>	<code>input:invalid</code>	seleciona todos os input inválidos
<code>:optional</code>	<code>input:optional</code>	seleciona todos os input <b>não</b> obrigatórios
<code>:required</code>	<code>input:required</code>	seleciona todos os input obrigatórios

- *Pseudo Class Selectors* específicos para formulários

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    input:disabled { background-color:lightcoral}
    input:focus {background-color:lightgoldenrodyellow}
    input:invalid {background-color:lightgray}
    input:required {background-color:lightgreen}
  </style>
</head>
<body>
  <form>
    <input type="text" size="30" placeholder="First Name" disabled/> <br/><br/>
    <input type="text" size="30" placeholder="Last Name" autofocus/><br/><br/>
    <input type="email" size="30" placeholder="Email"/><br/><br/>
    <input type="password" size="30" placeholder="Password" required/><br/><br/>
    <input type="submit" value="sign up" />
  </form>
</body>
</html>
```

The image shows a visual representation of the HTML form defined in the code. It consists of four input fields and a submit button. The 'First Name' field is disabled and has a light coral background. The 'Last Name' field is focused and has a light goldenrod yellow background. The 'Email' field is invalid and has a light gray background. The 'Password' field is required and has a light green background. The submit button is labeled 'sign up'.

- Lista completa de *pseudo class selectors*: [http://www.w3schools.com/cssref/css\\_selectors.asp](http://www.w3schools.com/cssref/css_selectors.asp)

## Pseudo-Elements Selectors

## Seletor CSS

- Referem-se a elementos fictícios (não correspondem a elementos HTML), os quais são baseados na estrutura do documento.
- Utilizam uma notação “::” diferente da pseudo-classe “:”
  - ::first-line**
  - ::first-letter**
  - ::before**
    - Com base na propriedade **content** permite inserir conteúdo antes de um elemento
  - ::after**
    - Com base na propriedade **content** permite inserir conteúdo depois de um elemento
  - ::selection**
    - Parte de um elemento que é selecionada pelo utilizador.

## Seletor CSS

- ::before ::after + propriedade content**

**\*\* ANTES \*\* Pseudo elements (::before;::after) \*\* DEPOIS \*\***

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    p {color:gray;}
    span {color:orange;}
    p::before {content:"** ANTES **"}
    p::after {content:"** DEPOIS **"}
  </style>
</head>
<body>
  <p><span>Pseudo elements (::before;::after)</span></p>
</body>
</html>
```

### ■ **::selection**

- permite formatar o conteúdo que está a ser selecionado pelo utilizador

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>

  <style>
    ::selection {background-color:orange;color:white; }
  </style>
</head>
<body>
  <p><span> Formatação diferente para uma seleção de texto </span></p>
</body>
</html>
```

Formatação diferente para uma seleção de texto

## Conflitos de Formatação

## CSS - Conflitos de Formatação

- Tipos de Conflito
  - Seletores iguais
    - Ordem pela qual são definidos
  - Seletores Diferentes
    - Especificidade
  - Herança

## CSS - Conflitos de Formatação

- Seletores iguais
  - Ordem pela qual são definidos
    - *last one listed wins*: **prevalece a ultima definição** para **um determinado seletor**

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8" />
  <title> External Style Sheet</title>
  <style>
    h1{color:#F00;} /*red*/
    h1{color:#00F;} /*blue*/
  </style>
</head>

<body>
  <h1> Resolver Conflitos de Formatação </h1>
</body>
</html>
```

?

**Resolver Conflitos de Formatação**

## CSS - Conflitos de Formatação

### ■ Seletores iguais

#### ■ Ordem pela qual são definidos

- Um ficheiro externo foi incorporado **depois** da definição de um *embedded style*,

```
<style> ... </style>
<link ...>
```

em caso de conflito **para o mesmo seletor**, prevalece a formatação estabelecida no **ficheiro externo**.

- Um ficheiro externo foi incorporado **antes** da definição de um *embedded style*,

```
<link ...>
<style> ... </style>
```

em caso de conflito **para o mesmo seletor**, prevalece a formatação estabelecida no ***embedded style***.

## CSS - Conflitos de Formatação

### ■ Conflitos

#### ■ Seletores Iguais: Prioridade ?

- Critério: “Ordem pela qual são definidos”

1. Regra assinalada como **!important** [mais prioritário]

```
<style>
  h1{color:#F00 !important;}
</style>
```

#### **!important**

Prevalece sobre todas as formatações  
É a **exceção** à regra da “ordem pela qual são definidos”

2. *Inline style* (atributo **style** na *opening tag*)

3. *Embedded Style Sheet* ( **<style>...</style>** )

4. *External Style Sheet* ( **<link .../>**; **@import** )

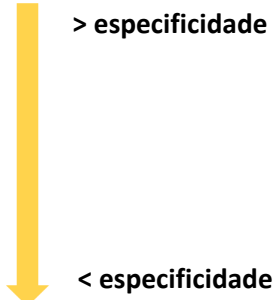
5. Definições por defeito do *browser* [menos prioritário]

Considera que o *embedded style* é declarado **após** a ligação ao ficheiro externo; caso contrário prevaleceria a declaração no ficheiro externo.

## CSS - Conflitos de Formatação

### ■ Especificidade

- Os seletores **mais específicos tem mais peso** na resolução de conflitos de formatação:

1. ID Selectors (mais específico / mais prioritário)
  2. Class Selectors; Attribute Selectors
  3. Contextual Selectors (apenas com elementos)
  4. Individual Element Selectors
- 

- Entre seletores com **a mesma especificidade prevalece** a ultima declaração.

## CSS - Conflitos de Formatação

### ■ Especificidade

- O browser **atribui pesos específicos** de acordo com o tipo de seletor
- Como calcular?
  - **d**: seletores de elemento e pseudo-elemento (**1 ponto**)
  - **c**: seletores de class, atributo e *pseudo-class* (**10 pontos**)
  - **b**: seletor de ID (**100 pontos**)
  - **a**: *inline style* (**1000 pontos**)

**2** → `<style>`  
    `body h1{color:red;}`  
**1** → `h1{color:blue;}`  
    `</style>`  
    `</head>`

**Resolução de conflitos de formatação**

`<body>`  
    `<h1> Resolução de conflitos de formatação </h1>`



## CSS - Conflitos de Formatação

### ■ Conflitos de Formatação - Prioridades na aplicação de estilos

#### ■ Especificidade

- Como calcular?

a	b	c	d

```
<style>
13  body div.especifica h1{color:darkolivegreen}
2   body h1{color:red}
1   h1 {color:blue}
</style>
</head>
<body>
  <div class="especifica">
    <h1 id="maisEspecifico">Resolução Conflitos de Formatação </h1>
  </div>
```

## Resolução Conflitos de Formatação

## CSS - Conflitos de Formatação

### ■ Conflitos de Formatação - Prioridades na aplicação de estilos

#### ■ Especificidade

a	b	c	d

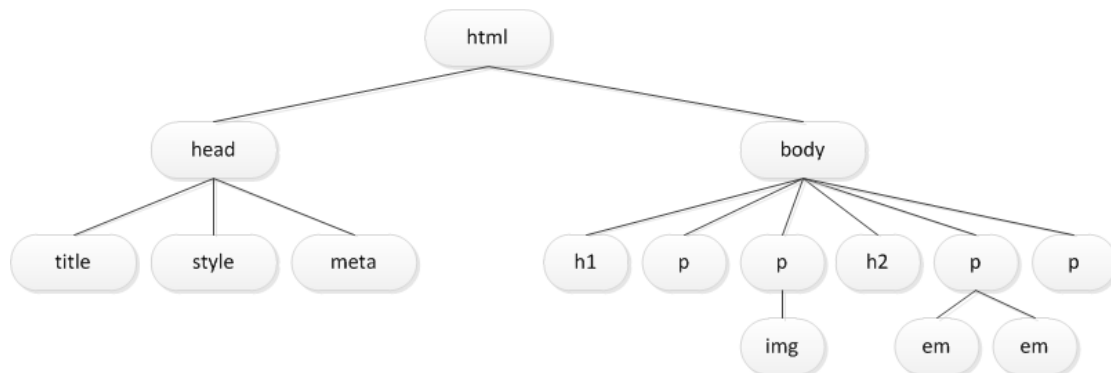
```
<style>
100 → #maisEspecifico{color:darkorange}
      body div.especifica h1{color:darkolivegreen}
      body h1{color:red}
      h1 {color:blue}
</style>
</head>
<body>
  <div class="especifica">
    <h1 id="maisEspecifico">Resolução Conflitos de Formatação </h1>
  </div>
```

## Resolução Conflitos de Formatação

## CSS - Conflitos de Formatação

### ■ Herança

- Documentos HTML tem uma estrutura hierárquica implícita



- body head filhos de html
- h1 p em h2 img descendentes de body
- ...

## CSS - Conflitos de Formatação

### ■ Herança

- **Algumas** propriedades de **alguns** elementos HTML são herdadas
  - Geralmente as propriedades relacionadas com o estilo do texto são herdadas

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    p {color:gray;}
  </style>
</head>
<body>
  <p> Conceito Herança <span> formatação herdada de p </span></p>
</body>
</html>
```

Conceito Herança formatação herdada de p

O elemento `<span>` é formatado por herança uma vez que não foi formatado diretamente.  
Na realidade, apenas foi formatado o elemento pai `<p>`.

## ■ Herança

- Só é aplicada **caso não seja definido** o estilo do elemento.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <style>
    .p1 {color:gray;}
  </style>
</head>
<body>
  <p class="p1"> Conceito Herança <span> formatação herdada de p </span></p>
</body>
</html>
```

Conceito Herança formatação herdada de p

- Caso o elemento seja diretamente formatado, a herança não se aplica: **prevalece sempre a formatação direta do elemento**

Conceito Herança formatação herdada de p

```
<style>
  .p1 {color:gray;}
  span {color:orange;}
</style>
```

- Este funcionamento aplica-se sempre, mesmo que o seletor aplicado ao elemento tenha menor especificidade do que a especificidade do seletor aplicado ao elemento pai