



# Programação para Internet

---

Módulo 2 – Parte 1

Introdução à Linguagem CSS

Prof. Dr. Daniel A. Furtado - FACOM/UFU

Conteúdo protegido por direito autoral, nos termos da Lei nº 9 610/98

A cópia, reprodução ou apropriação deste material, total ou parcialmente, é proibida pelo autor

# Conteúdo da Aula

1. Introdução à CSS
2. Seletores
3. Unidades absolutas e relativas
4. Propriedades de ajuste de fonte e texto

# O que é CSS?

- Linguagem de estilos para definir aspectos de apresentação
- **Cascading Style Sheets** - Folhas de Estilo em Cascata
- HTML se preocupa com o conteúdo da página
- CSS se preocupa com os estilos de apresentação desse conteúdo
- Separação entre conteúdo e apresentação gráfica
- Desenvolvida pelo **CSS Working Group** no **W3C**

# Por que estudar CSS?

- Tecnologia de base
- Solidez e maturidade
- Independência dos frameworks
- Maior controle dos estilos e layouts
- Apenas o código necessário



Prof. Daniel A. Furtado



- [INÍCIO](#)
- [ENSINO](#)
- [PROJETOS](#)
- [PUBLICAÇÕES](#)

### Programação para Internet

A disciplina tem como objetivo capacitar o aluno para o desenvolvimento de aplicações Web utilizando as tecnologias de base, com foco no desenvolvimento do front-end e na programação direta do back-end, incluindo acesso a banco de dados.

Os objetivos específicos incluem: 1) discutir o funcionamento de sistemas Web e os protocolos envolvidos; 2) discutir o paradigma da programação para a Web e 3) desenvolver interfaces gráficas para a Web; 4) desenvolver websites dinâmicos e interativos através da programação direta do back-end e 5) utilizar conceitos e tecnologias para acesso a banco de dados em sistemas Web.

### Sistema de Avaliação

Serão aplicadas três avaliações práticas, um projeto de implementação e vários testes de aula. As avaliações práticas devem ser realizadas em horário de aula, sob supervisão do professor. O projeto de implementação deverá ser apresentado pela equipe no final do semestre letivo, conforme cronograma disponibilizado pelo professor.

### Sistemas de Banco de Dados

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Magnam delectus impedit at molestias commodi autem odit maxime necessitatibus nam, accusantium velit natus, minus veritatis. Ex repellendus earum totam similique porro. Magnam delectus impedit at molestias commodi autem odit maxime necessitatibus nam, accusantium velit natus, minus veritatis.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Et quidem illum temporibus consequatur in sequi consectetur minus nobis. Provident maiores repellendus architecto aliquid quos quas magnam nobis. Perspiciatis, molestias ratione.

### Tutoriais

- [Modelagem](#)
- [DERs](#)
- [SBDs](#)
- [Mozilla Dev](#)
- [WHATWG](#)
- [W3C](#)
- [W3Schools](#)
- [DERs](#)
- [SBDs](#)
- [Mozilla Dev](#)
- [WHATWG](#)

Prof. Daniel A. Furtado

INÍCIO   ENSINO   PROJETOS   PUBLICAÇÕES

### Programação para Internet

A disciplina tem como objetivo capacitar o aluno para o desenvolvimento de aplicações Web utilizando as tecnologias de base, com foco no desenvolvimento do front-end e na programação direta do back-end, incluindo acesso a banco de dados.

Os objetivos específicos incluem: 1) discutir o funcionamento de sistemas Web e os protocolos envolvidos; 2) discutir o paradigma da programação para a Web e 3) desenvolver interfaces gráficas para a Web; 4) desenvolver websites dinâmicos e interativos através da programação direta do back-end e 5) utilizar conceitos e tecnologias para acesso a banco de dados em sistemas Web.

### Sistema de Avaliação

Serão aplicadas três avaliações práticas, um projeto de implementação e vários testes de aula. As avaliações práticas devem ser realizadas em horário de aula, sob supervisão do professor. O projeto de implementação deverá ser apresentado pela equipe no final do semestre letivo, conforme cronograma disponibilizado pelo professor.

### Sistemas de Banco de Dados

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Magnam delectus impedit at molestias commodi autem odit maxime necessitatibus nam, accusantium velit natus, minus veritatis. Ex repellendus earum totam similique porro. Magnam delectus impedit at molestias commodi autem odit maxime necessitatibus nam, accusantium velit natus, minus veritatis.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Et quidem illum temporibus consequatur in sequi consectetur minus nobis. Provident maiores repellendus architecto aliquid quos quas magnam nobis. Perspiciatis, molestias ratione.

Tutoriais

- [Modelagem](#)
- [DERs](#)
- [SBDs](#)
- [Mozilla Dev](#)
- [WHATWG](#)
- [W3C](#)
- [W3Schools](#)
- [DERs](#)
- [SBDs](#)
- [Mozilla Dev](#)
- [WHATWG](#)

© Copyright 2020. Todos os direitos reservados.

# Três Formas de Inserir CSS

## 1. Embutido na linha (*inline*)

- Atributo *style*
- O uso deve ser evitado (difícil manutenção)

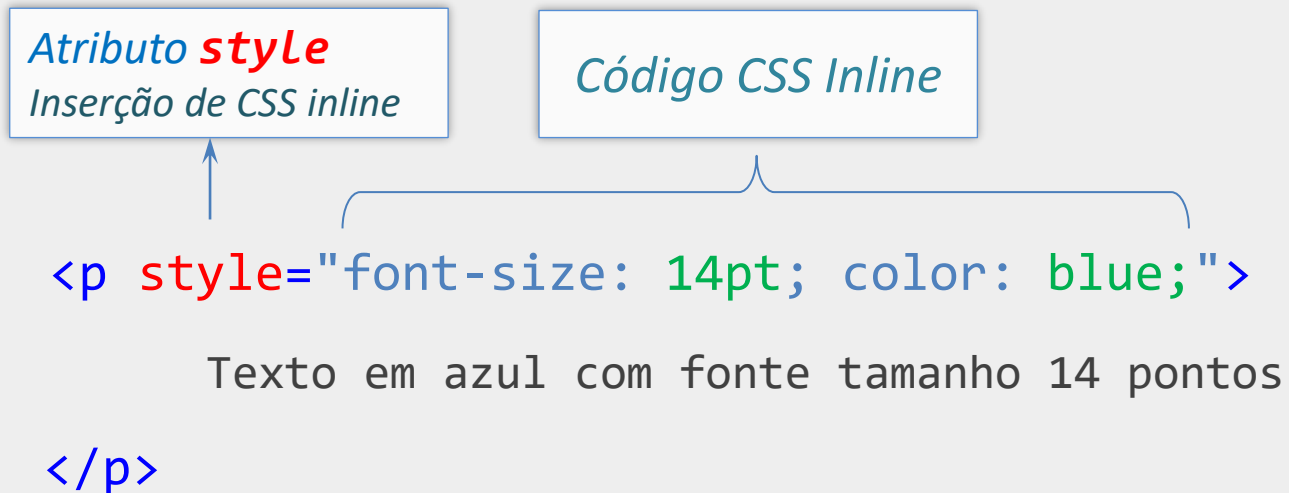
## 2. Folha de estilos embutida

- Utiliza o elemento `<style>` dentro do `<head>`
- Estilos específicos da página, não compartilhados

## 3. Folha de estilos em arquivo separado

- Utiliza o elemento `<link>` para referenciar arquivo CSS
- Várias páginas podem utilizar o mesmo código CSS
- Melhor separação entre conteúdo e estilos

# CSS Embutido na Linha (*inline*)



- O código CSS afetará apenas o elemento em questão

# CSS em Folha de Estilos Embutida

```
<html>

  <head>

    <style>
      p {
        font-size: 14pt;
        color: blue;
      }
    </style>

  </head>

  <body>
    <p> ... </p>
  </body>

</html>
```

*Código CSS embutido dentro do elemento `<style>`, no cabeçalho do documento*



# CSS em Folha de Estilos Separada

## Arquivo HTML

```
<html>

  <head>

    <link rel="stylesheet" href="style.css">

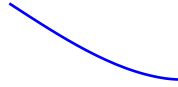
  </head>

  <body>

    <p> ... </p>

  </body>

</html>
```

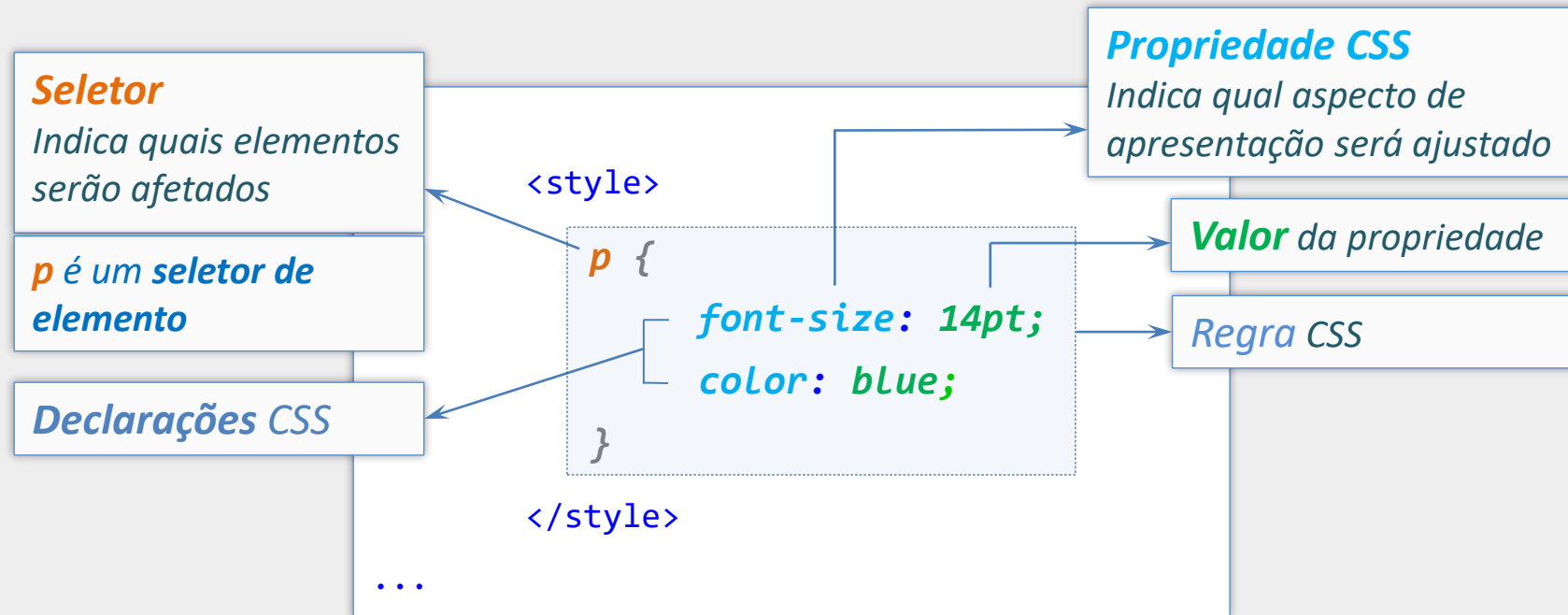


## Arquivo CSS

```
/* Arquivo style.css */

p {
  font-size: 14pt;
  color: blue;
}
```

# Regra, Seletor e Propriedades



*Propriedades e valores CSS são case-sensitive*

# Múltiplas Regras e Seletores

...

```
<style>
```

```
body {  
    background-color: gray;  
}
```

*Afetar  o corpo do documento HTML*

```
p {  
    font-size: 20pt;  
    color: blue;  
}
```

*Afetar  todos os elementos **p***

```
h1 {  
    font-family: Verdana;  
}
```

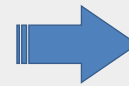
*Afetar  todos os t tulos **h1***

```
</style>
```

...

# Agrupando Seletores

```
...  
<style>  
    h1 {  
        font-family: Verdana;  
        color: darkblue;  
    }  
    h2 {  
        font-family: Verdana;  
        color: darkblue;  
    }  
    h3 {  
        font-family: Verdana;  
        color: darkblue;  
    }  
</style>  
...
```



```
...  
<style>  
    h1, h2, h3 {  
        font-family: Verdana;  
        color: darkblue;  
    }  
</style>  
...
```

# Seletor de ID (*#idDoElemento*)

Quando se deseja aplicar estilos a apenas  
**um elemento em particular**

# Seletor de ID (*#idDoElemento*)

```
<style>
```

```
#par1 {  
  text-align: center;  
  color: red;  
}
```

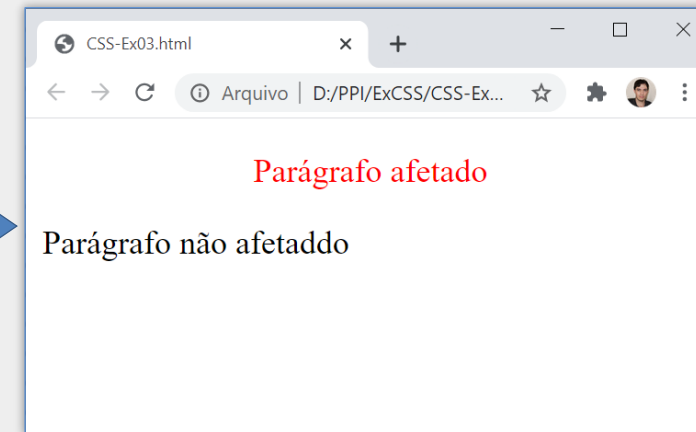
```
</style>
```

```
<body>
```

```
<p id="par1">Parágrafo afetado</p>
```

```
<p>Parágrafo não afetado</p>
```

```
</body>
```



*Afeta apenas o elemento que tem aquele valor de **id***

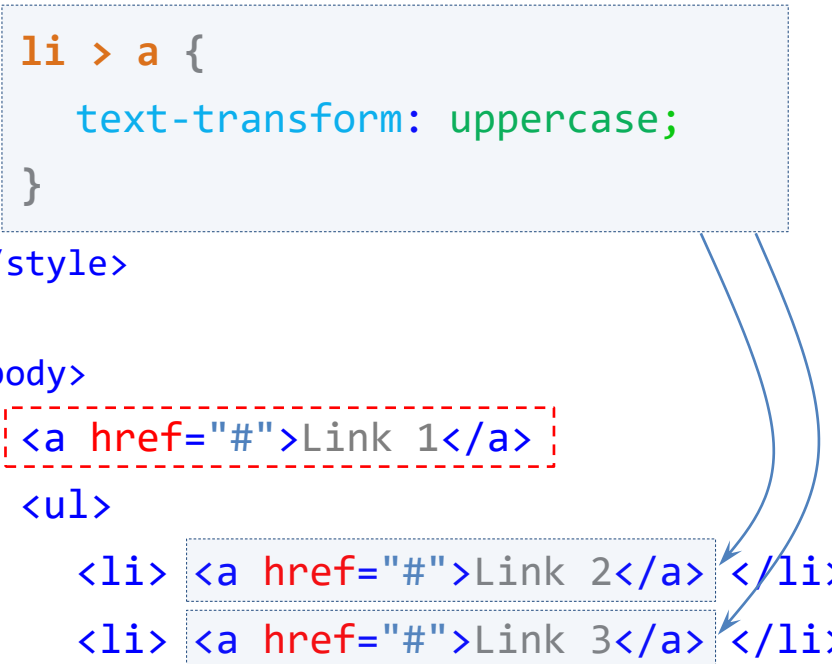
# Seletor de Filho ( $x > y$ )

Afeta os elementos  $y$  **que são filhos** de elementos  $x$

# Seletor de Filho (*x* > *y*)

```
...
<style>
  li > a {
    text-transform: uppercase;
  }
</style>
...
<body>
  <a href="#">Link 1</a>
  <ul>
    <li> <a href="#">Link 2</a> </li>
    <li> <a href="#">Link 3</a> </li>
  </ul>
...

```



*Afetar  todos os elementos <a>  
que s o filhos de elementos <li>  
(Link 2 e Link 3)*



# Seletor de Descendente (*x* *y*)

Afeta os elementos *<y>* **que estão dentro** de elementos *<x>*, mesmo que tenha outros elementos aninhados entre eles

# Seletor de Descendente (*x* *y*)

```
...


Afetar  todos os elementos <a> que  
est o dentro de elementos <div>,  
direta ou indiretamente  
(Link 2 e Link 3)



Programa o para Internet



Prof. Dr. Daniel A. Furtado - Proibida c pia, apropria o ou uso sem autoriza o de qualquer parte deste material - Lei n  9 610/98



18

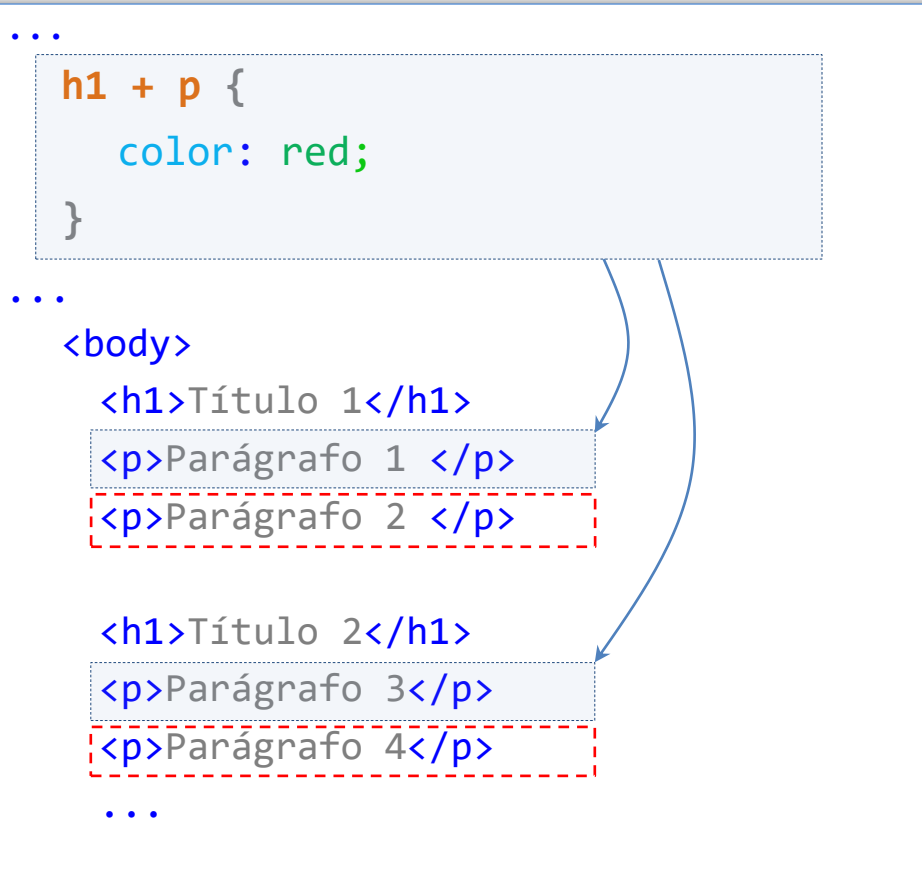

```

# Seletor de Irmão Adjacente ( $x + y$ )

Afeta o elemento  $\langle y \rangle$  caso ele esteja **imediatamente depois** de  $\langle x \rangle$

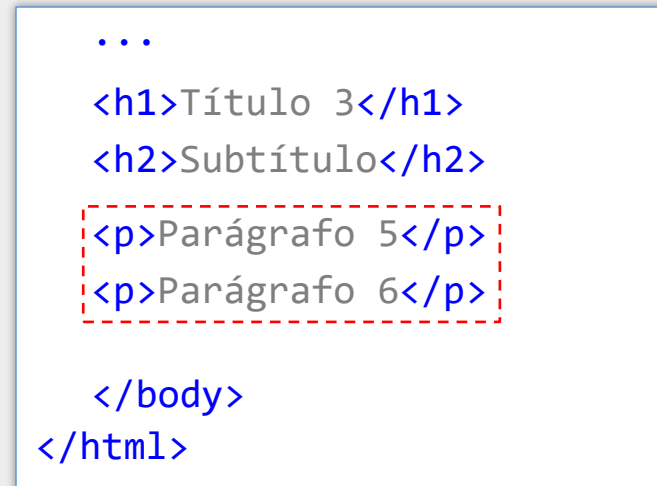
# Seletor de Irmão Adjacente (*x* + *y*)

```
...  
h1 + p {  
  color: red;  
}  
...  
<body>  
  <h1>Título 1</h1>  
  <p>Parágrafo 1 </p>  
  <p>Parágrafo 2 </p>  
  
  <h1>Título 2</h1>  
  <p>Parágrafo 3</p>  
  <p>Parágrafo 4</p>  
  ...
```



*Afetar*á todo parágrafo que está **imediatamente depois** de um `<h1>`

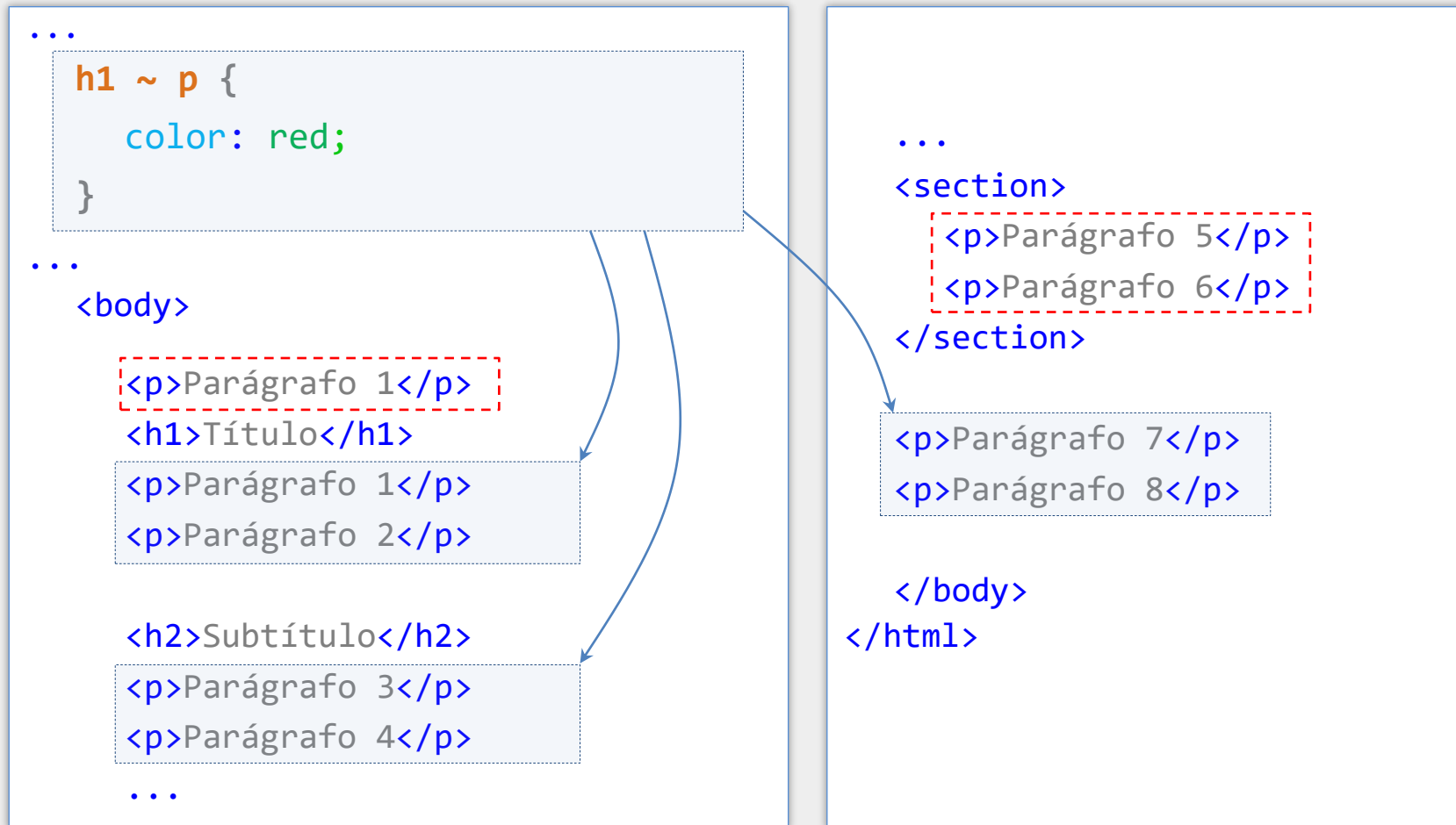
```
...  
<h1>Título 3</h1>  
<h2>Subtítulo</h2>  
  <p>Parágrafo 5</p>  
  <p>Parágrafo 6</p>  
  
</body>  
</html>
```



# Seletor de Irmão Geral ( $x \sim y$ )

Afeta todos os elementos  $\langle y \rangle$  que são **irmãos** de  $\langle x \rangle$  e aparecem **depois** de  $\langle x \rangle$

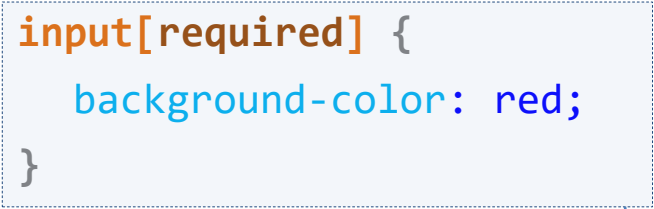
# Seletor de Irmão Geral ( $x \sim y$ )



# Seletor de Atributo *x[atrib]*

Seleciona elementos de acordo com alguma condição envolvendo seus atributos

# Seletor de Atributo *x[atrib]*

```
...


A light blue box with a dashed border contains the CSS selector input[required] { background-color: red; }. A blue arrow points from this box to the <input type="text" required> element in the HTML code below. The <input type="text"> element in the HTML code is enclosed in a red dashed box.


```

*Seleciona os elementos `<input>`  
que possuem o atributo `required`*



# Seletor de Atributo *x[atrib]*

```
...  
input[type="text"] {  
    margin: 10px;  
}  
...  
<form>  
    Nome: <input type="text">  
    Cidade: <input type="text">  
    Senha: <input type="password">  
</form>  
...
```

*Seleciona os elementos <input>  
cujo atributo **type** tem o valor text*

*Há muitas outras possibilidades de  
seleção envolvendo atributos.*

# Seletor de Classe (*.nomeClasse*)

- Situações onde se pretende aplicar os estilos mais de uma vez
- Cria-se uma classe com estilos pré-definidos
- Aplica-se os estilos da classe onde for necessário

# Seletor de Classe (*.nomeClasse*)

Criação da classe de estilos

```
.nomeDaClasse {  
    /* declarações CSS */;  
}
```

Aplicação da classe de estilos

```
...  
<elemento class="nomeDaClasse">  
...
```

# Seletor de Classe (*.nomeClasse*)

...

```
.destacado {  
  color: blue;  
  text-transform: uppercase;  
}
```

...

```
<h1 class="destacado">Título 1</h1>
```

```
<p>Parágrafo 1 </p>
```

```
<p class="destacado">Parágrafo 2 </p>
```

```
<p>Parágrafo 3 </p>
```

```
<p>Parágrafo 4 </p>
```

```
<p class="destacado">Parágrafo 5 </p>
```

...


*Afetar  o t tulo 1  
e os par grafos 2 e 5*

# Seletor de Classe (*x.nomeClasse*)

Define uma classe de estilos para ser utilizada  
**apenas no elemento** `<x>`

# Seletor de Classe (*x.nomeClasse*)

```
...  
p.destacado {  
  color: blue;  
  text-transform: uppercase;  
}  
...  
  
[<h1 class="destacado">Título 1</h1>  
<p>Parágrafo 1 </p>  
<p class="destacado">Parágrafo 2 </p>  
...]
```



*Afetar*á os **parágrafos** que fizerem uso da classe **destacado**

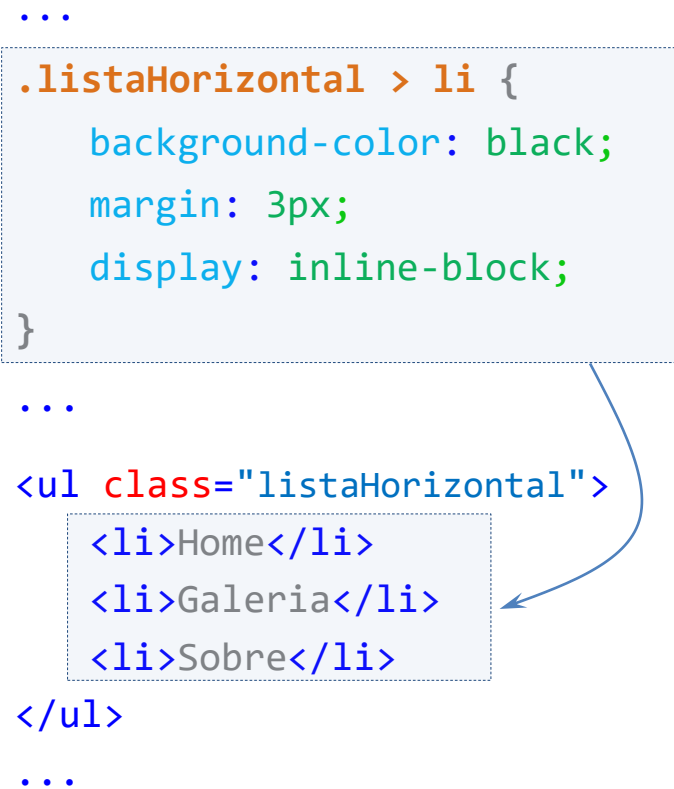
# Seletor de Classe (*x.nomeClasse*)



# Seletor de Classe (*.nomeClasse*)

- É possível utilizar outros seletores em conjunto com classes

```
...  
...  
.listaHorizontal > li {  
    background-color: black;  
    margin: 3px;  
    display: inline-block;  
}  
...  
<ul class="listaHorizontal">  
    <li>Home</li>  
    <li>Galeria</li>  
    <li>Sobre</li>  
</ul>  
...
```



*Afetará os <li> que são  
filhos de elementos que  
utilizam a classe  
*.listaHorizontal**



# Referenciando Múltiplas Classes

```
...  
.destacado {  
    color: blue;  
    font-style: italic;  
}  
  
.centralizado {  
    text-align: center;  
}  
  
...  
<h1 class="destacado centralizado">Título 1</h1>  
...
```

Se houver repetição de propriedades, prevalecerão aquelas referenciadas por último (nesta situação)

# Propriedades CSS

---

# Propriedades de Ajuste de Fonte

Propriedade	Descrição	Exemplos de Valores
<b>font-family</b>	Define a fonte	<i>"Times New Roman", Times, serif;</i>
<b>font-style</b>	Estilo da fonte	<i>normal, italic, oblique, etc.</i>
<b>font-size</b>	Tamanho da fonte	<i>12pt, 20px, 2em, 2rem, etc.</i>
<b>font-weight</b>	Define a espessura da letra	<i>lighter, normal, bold, bolder</i> <i>ou valor entre 100 e 900</i>
<b>font-variant</b>	Define variantes da fonte	<i>normal, small-caps</i> EXEMPLO DE SMALL-CAPS
<b>font-stretch</b>	Estica ou comprime a fonte Nem toda fonte é compatível	<i>condensed, normal, expanded</i>
<b>line-height</b>	Espaçamento entre linhas	<i>16pt, 20px, 1.5</i>

# Propriedades de Ajuste de Texto

Propriedade	Descrição	Exemplos de Valores
<b>text-align</b>	Alinhamento horizontal	<i>left, center, right, justify</i>
<b>text-decoration</b>	Decoração adicional	<i>none, underline, overline, line-through</i>
<b>text-indent</b>	Recuo de 1ª linha	<i>30px</i>
<b>text-transform</b>	Controle de maiúsculas e minúsculas	<i>uppercase, lowercase, capitalize</i>
<b>vertical-align</b>	Alinhamento vertical	<i>top, middle, bottom</i>
<b>color</b>	Define a cor do texto	<i>red, gray, #FAFAFA, #DDD</i>

# Ajuste de Cor

- Nome da cor
  - `blue`, `darkblue`, `lightblue`, etc.
- Valor RGB
  - `rgb(0, 120, 255)`
  - `rgba(0, 120, 255, 128)` (50% translúcido)
- Código Hexadecimal
  - Notação com 6 dígitos. Exemplo: `#FF0000`
  - Notação com 3 dígitos. Exemplo: `#AF5` (equivalente a `#A AFF55`)
  - Com transparência. Exemplo: `#FF000080` (50% translúcido)
- Cor HSL (matiz, saturação, luminosidade)
  - `hsl(m, s, l)`
  - `m`: 0-360; `s`: 0-100%, `l`: 0-100%;
- Recurso dos editores para definição de cores

# Ajuste de Cor - Exemplo

Algumas formas diferentes de definir exatamente a mesma cor

```
p { color: blue; }  
p { color: #0000FF; }  
p { color: #00F; }  
p { color: rgb(0, 0, 255); }  
p { color: hsl(240, 100%, 50%); }
```

# Propriedade *font-family*

```
h1 {  
    font-family: "Times New Roman", Georgia, serif;  
}
```

- Possibilidade de indicar mais de um valor
- O próximo é utilizado em caso de ausência do anterior
- Recomenda-se encerrar com nome de **família genérica**

# Alguns Tipos de Fontes



## **Fonte Sans-serif**

Sem prolongamentos

Ex.: Aria, Verdana



## **Fonte Serif**

Com prolongamentos

Ex.: Times New Roman



## **Fonte Monospace**

Letras com mesma largura de exibição

Ex.: Courier New, Consolas



# Tamanhos de Fonte

## ■ Unidades de Tamanho Absoluto

- Texto sempre do mesmo tamanho
- Não afetado pelas configurações de tamanho de fonte do navegador
- Não afetado pelo tamanho de fonte do elemento pai
- Exemplos
  - Valor em **pixels** (px)
  - Valor em **pontos** (pt)
  - Palavras-chave: *x-small*, *small*, *medium*, *large*, *x-large*, *xx-large*, *etc.*
- Devem ser utilizados com cautela por desprezar os tamanhos de fonte do usuário

# Tamanhos de Fonte

## ■ Unidades de Tamanho Relativo

- Pode depender das configurações de tamanho de fonte do navegador
- Pode depender do tamanho da fonte do elemento pai
- Exemplos: `em`, `rem`, `%`, `larger`, `smaller`

# Tamanhos de Fonte

- Tamanho Relativo *em*
  - Relativo ao tamanho da fonte corrente (do elemento pai)
  - *2em* = dobro da fonte corrente
- Tamanho Relativo *rem*
  - Relativo ao tamanho da fonte do elemento raiz (html)
- Outros Relativos
  - *larger*, *smaller*, %
  - Relativo ao tamanho da fonte do elemento pai (ou fonte do navegador)

# Exemplo de Tamanho *em*

```
.medio {  
  font-size: 16px;  
}  
  
.grande {  
  font-size: 2em;  
}
```

```
<main class="medio">  
  Texto 16px  
  <section class="grande">  
    Texto 32px  
    <article class="grande">  
      Texto 64px  
    </article>  
  </section>  
</main>
```

# Exemplo de Tamanho *rem*

```
html {  
  font-size: 14px;  
}  
  
.medio {  
  font-size: 16px;  
}  
  
.grande {  
  font-size: 2rem;  
}
```

```
<main>  
  Texto 14px  
  <section class="medio">  
    Texto 16px  
    <article class="grande">  
      Texto 28px  
      <div class="grande">  
        Texto 28px  
      </div>  
    </article>  
  </section>  
</main>
```

# Exemplo de Tamanho *rem*

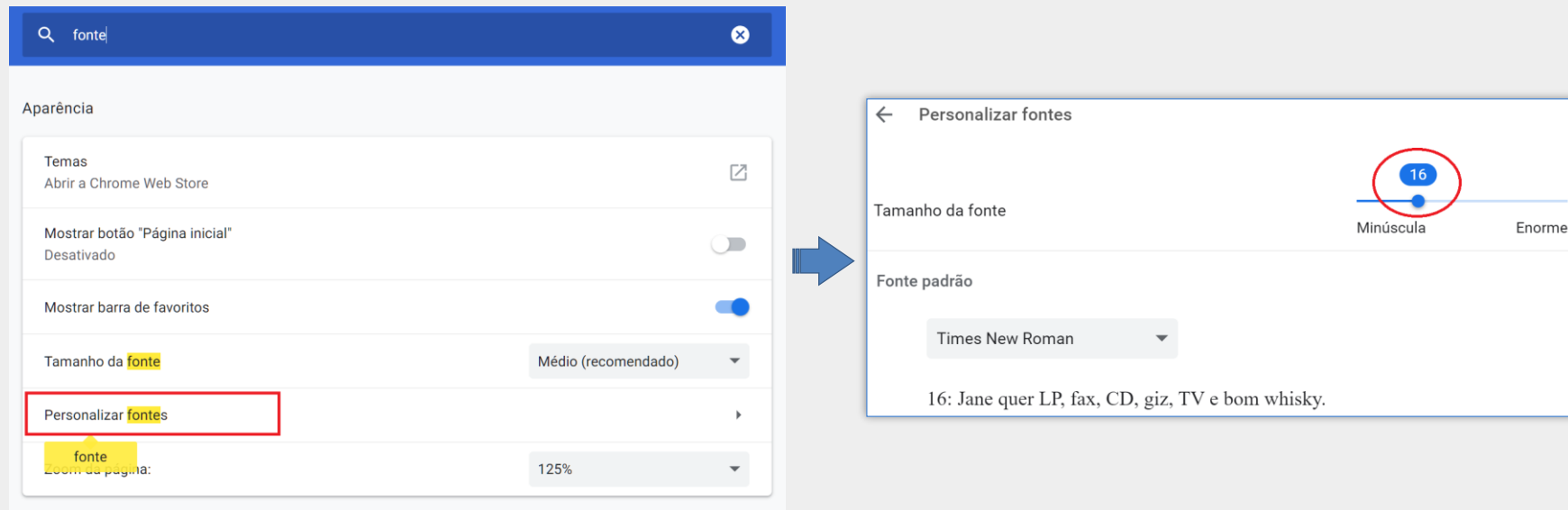
Considere que o tamanho da **fonte padrão** definida no **navegador** seja 18px

```
.medio {  
  font-size: 16px;  
}  
  
.grande {  
  font-size: 2rem;  
}
```

```
<main>  
  Texto 18px  
  <section class="medio">  
    Texto 16px  
    <article class="grande">  
      Texto 36px  
      <div class="grande">  
        Texto 36px  
      </div>  
    </article>  
  </section>  
</main>
```

# Tamanho Padrão de Fonte no Navegador

No Google Chrome, acesse *Configurações*, digite *fonte* e escolha *Personalizar fontes*



# Propriedade *font*

- Propriedade abreviada (*shorthand property*)
- Permite definir em uma única linha as propriedades constituintes
  - `font-family`
  - `font-size`
  - `font-style`
  - `font-weight`
  - `font-stretch`
  - `font-variant`
  - `line-height`



# Propriedade *font*

## Exemplos de uso

- `font: 18px Verdana;`
- `font: bold 16px Georgia;`
- `font: italic 14px /1.5 Consolas;`

# Propriedade *font*

`font:`   *italic*   *small-caps*   *condensed*   *bold*   <sup>obrigatório</sup> *16px*   */1.5*   <sup>obrigatório</sup> *Arial*

*font-style*   *font-variant*   *font-stretch*   *font-weight*   *font-size*   *line-height*   *font-family*

- Apenas os valores de *font-size* e *font-family* são obrigatórios
- *font-family* deve ser o último valor
- *font-style*, *font-variant* e *font-weight* devem vir antes de *font-size*
- *line-height* deve vir logo depois de *font-size*, acompanhado de /
- A omissão de uma propriedade constituinte retorna seu valor ao valor inicial

# Referências

- [www.w3.org/Style/CSS/](http://www.w3.org/Style/CSS/)
- [developer.mozilla.org/en-US/docs/Web/CSS](http://developer.mozilla.org/en-US/docs/Web/CSS)
- [www.w3.org/Style/CSS/learning](http://www.w3.org/Style/CSS/learning)
- **HTML and CSS: Design and Build Websites**, Jon Duckett.