

**CSS**

# Cascading Style Sheets

- “folhas de estilo em cascatas”;
- linguagem de estilização (**não é linguagem de programação**);
- criado para resolver os problemas de estilização do HTML (refatoração);
- utilizado para estilizar páginas construídas com HTML;
- possui uma grande variedade de atributos e valores;
- versão atual: CSS3;

# Melhorias do CSS3

- CSS lançado em 1996, CSS2 em 1998, CSS3 em 1999;
- novas opções de estilização;
- cantos arredondados, sombras, gradientes, animações, efeitos de transição;
- seletores complexos;
- pseudo-elementos;

# CSS *inline*

- utilizado para aplicar um estilo a um único elemento;
- definido através do atributo `style`;

# CSS *inline*

Exemplo:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <body>
```

```
    <h1 style="color: blue; text-align: center;">This is a heading.</h1>
```

```
    <p style="color: red;">This is a paragraph.</p>
```

```
  </body>
```

```
</html>
```

# CSS interno

- utilizado para aplicar um estilo a vários elementos dentro de uma página;
- definido dentro da tag `<style>` dentro do `<head>`;

# CSS interno

Exemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h1 { color: blue; text-align: center; }
      p { color: red; }
    </style>
  </head>
  <body>
    <h1>This is a heading.</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

# CSS externo

- pode ser utilizado para aplicar um estilo a várias páginas, pois pode ser reaproveitado entre documentos;
- o arquivo HTML fica menor, pois existe (ao menos) um arquivo CSS;
- definido através da tag `<link>` dentro do `<head>`;



# CSS externo

Exemplo (HTML):

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>This is a heading.</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

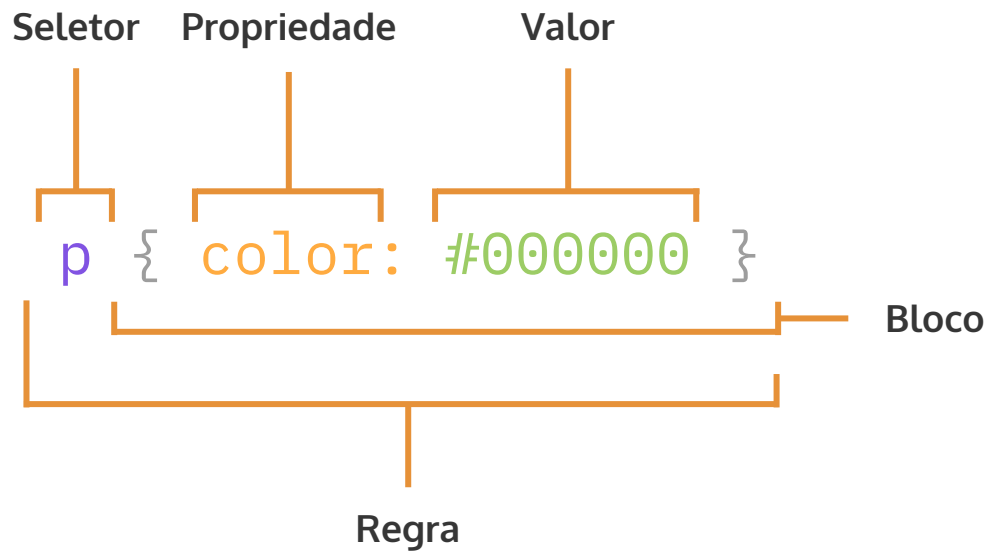
# CSS externo

Exemplo (CSS):

```
h1 {  
    color: blue;  
    text-align: center;  
}
```

```
p {  
    color: red;  
}
```

# Sintaxe do CSS



# Sintaxe do CSS

- uma regra de CSS consiste em um seletor e um bloco de declaração:
  - **seletor:** indica o elemento HTML que será estilizado;
  - **bloco de declaração:** contém uma ou mais declarações, compostas por propriedade e valor;
  - **propriedade:** define o que será estilizado;
  - **valor:** define como será estilizado;
- cada bloco inicia e termina com "{ }" (chaves);
- cada declaração é separada por ";" (ponto e vírgula);
- entre a propriedade e o valor, deve ter um ":" (dois pontos);

# Seletores simples

## Tag:

- seleciona todos elementos deste tipo de *tag*;
- sintaxe: `nome-da-tag`
- exemplo: `p { color: #000000 }`

# Seletores simples

## Classe

- seleciona os elementos que possuem esta classe como atributo;
- sintaxe: `.nome-da-classe`
- exemplo: `.paragraph { color: #000000 }`

# Seletores simples

## ID

- selecione elementos baseados no valor do atributo id (**deve existir apenas um elemento com um determinado ID no documento**);
- sintaxe: `#nome-do-id`
- exemplo: `#unique-paragraph { color: #000000 }`

# Seletores simples

## Universal

- seleciona todos os elementos;
- sintaxe: \*
- exemplo: \* { color: #000000 }



# Seletores simples

## Atributo

- seleciona elementos baseado no valor de seus atributos;
- sintaxe: `[atrib]`, `[atrib=valor]`, `[atrib~=valor]`, `[atrib|=valor]`,  
`[atrib^=valor]`, `[atrib$=valor]`, `[atrib*=valor]`
- exemplo: `[atrib=valor] { color: #000000 }`

# Seletores combinadores

## Descendentes

- seleciona elementos que são filhos (não necessariamente diretos) do elemento especificado;
- sintaxe: `a b`
- exemplo: `div p { color: #000000 }`

# Seletores combinadores

## Filhos

- seleciona elementos que são filhos diretos do elemento especificado;
- sintaxe: `a > b`
- exemplo: `div > p { color: #000000 }`

# Seletores combinadores

## Irmãos adjacentes

- seleciona elementos que seguem imediatamente o elemento especificado;
- sintaxe: `a + b`
- exemplo: `div + p { color: #000000 }`

# Seletores combinadores

## Irmãos gerais

- seleciona elementos que seguem (não necessariamente imediatamente) o elemento especificado dentro de um mesmo pai;
- sintaxe: `a ~ b`
- exemplo: `div ~ p { color: #000000 }`

# Seletores pseudo-classes

- pseudo-classes permitem selecionar elementos baseados em informações que não estão contidas na árvore do documento, como um estado especial ou algo particularmente complexo de extrair;
  - estilizar um elemento quando o mouse passar por cima;
  - estilizar de forma diferente links que já foram e não foram visitados;
- sintaxe: `a:pseudo-class`
- exemplo: `a:hover { color: #000000 }`

# Seletores pseudo-elementos

- pseudo-elementos são abstrações da árvore que representam entidades além do HTML, como o marcador de uma lista;
  - estilizar a primeira letra ou linha de um elemento;
  - inserir um conteúdo antes ou depois de um elemento;
- sintaxe: `a::pseudo-element`
- exemplo: `a::first-letter { color: #000000 }`

# Agrupamento de seletores

- é possível reutilizar um bloco de estilização para vários seletores;
- cada seletor deve ser separado por uma "," (vírgula);
- sintaxe: `a, b, c`
- exemplo: `p, .paragraph, #unique-paragraph { color: #000000 }`



# Ordem da cascata

- define qual estilo será utilizado quando mais de um estilo é aplicado ao mesmo elemento;
- *inline* > interno / externo > padrão do navegador;
- o *inline* possui a maior prioridade, enquanto o padrão do navegador possui a menor;
- interno e externo possuem a mesma prioridade (a ordem de declaração no `<head>` define qual irá prevalecer);

# Especificidade

- espécie de grade de pontuação que define qual estilo será aplicado a um determinado elemento quando há múltiplas declarações;
  - universal: 0 pontos;
  - elemento ou pseudo-elemento: 1 ponto;
  - classe ou pseudo-classe: 10 pontos;
  - id: 100 pontos;
  - inline: 1000 pontos;
- começa em 0, a soma mais alta “ganha” e define o estilo;
- quando o valor de especificidade é o mesmo, o último declarado “ganha”;
- o modificador `!important` pode sobrescrever a especificidade;

# Especificidade

Seletor	Especificidade	Cálculo
p	1	1
p.test	11	1 + 10
p#demo	101	1 + 100
<p style="color: pink">	1000	1000
#demo	100	100
.test	10	10
p.test1.test2	21	1 + 10 + 10
#navbar p#demo	201	100 + 1 + 100

# Propriedades comuns

color	cor de texto
background-color	cor de fundo
background-image	imagem de fundo
border	borda
border-radius	bordas arredondadas
margin	margem externa, para fora das bordas
padding	margem interna, para dentro das bordas
height	altura
width	largura
box-sizing	tamanho do box

# Propriedades comuns

<code>opacity</code>	opacidade/transparência
<code>text-align</code>	alinhamento do texto
<code>font-family</code>	família da fonte
<code>font-weight</code>	peso da fonte
<code>font-size</code>	tamanho da fonte
<code>line-height</code>	espaçamento entre linhas
<code>display</code>	comportamento de <i>display</i> do elemento
<code>position</code>	posicionamento
<code>z-index</code>	controle do eixo Z
<code>overflow</code>	comportamento de rolagem

# Propriedades comuns

<code>flex-direction</code>	direção do <i>flex container</i>
<code>flex-wrap</code>	envolvimento do <i>flex container</i>
<code>justify-content</code>	alinhamento horizontal dos itens do <i>flex container</i>
<code>align-items</code>	alinhamento vertical dos itens do <i>flex container</i>
<code>flex-grow</code>	define se os itens do <i>flex container</i> podem aumentar de tamanho
<code>flex-shrink</code>	define se os itens do <i>flex container</i> podem diminuir de tamanho
<code>flex-basis</code>	tamanho inicial de um item do <i>flex container</i>

# Links úteis

- [https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS\\_Selectors](https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS_Selectors)
- [https://www.w3schools.com/css/css\\_pseudo\\_classes.asp](https://www.w3schools.com/css/css_pseudo_classes.asp)
- [https://www.w3schools.com/css/css\\_pseudo\\_elements.asp](https://www.w3schools.com/css/css_pseudo_elements.asp)
- <https://www.w3schools.com/Css/default.asp>
- <https://www.w3schools.com/cssref/default.asp>