



**Objetivos:**

1. Compreender o CSS e sua importância para estilização de páginas web
2. Compreender a sintaxe básica
3. Aplicar estilos básicos nos elementos HTML

**Conteúdo Programático**

- Definição e Importância do CSS
- Sintaxe Básica
- Formas de Estilizar
- Seletores
- Propriedade Iniciais: Cores, Fontes e Textos.

**1 -) O que é CSS (*Cascading Style Sheet*)?**

É uma linguagem **de estilização** utilizada para definir a apresentação de uma página web, isto é, definir **como** os elementos HTML serão renderizados. O CSS permite separar a apresentação visual (**como os elementos devem aparecer?**) de um documento do conteúdo estrutural (**o que deve aparecer?**), permitindo que desenvolvedores páginas e aplicativos web elaborem layouts e designs mais reutilizáveis e flexíveis.

Utilizando CSS, conseguimos definir a cor, fonte, tamanho e posição de elementos em uma página web, além de controlar aspectos como margens, bordas, espaçamento e até mesmo animação entre elementos. O CSS é amplamente utilizado em conjunto com HTML e JavaScript para criar páginas web interativas e dinâmicas, e até mesmo jogos que executam inteiramente no navegador.

**2 -) Por que o CSS é importante para a criação de páginas web?**

O CSS é indispensável para a criação de páginas e aplicações web modernas. Utilizar o CSS traz benefícios para os(as) desenvolvedores(as) e usuários(as) finais. Para desenvolvedores (as), o CSS permite separar o design, isto é, as regras CSS do conteúdo (elementos HTML), tornando a manutenção mais fácil e eficiente. Com o uso do CSS, é possível criar um conjunto de estilos (regras) que pode ser aplicado a várias páginas diferentes, tornando a aparência consistente e evitando a repetição de código. Além disso, o CSS permite que o conteúdo seja apresentado de forma mais acessível e amigável para diferentes dispositivos e usuários, como por exemplo, tornando a página web responsiva para diferentes tamanhos de tela. É possível também adaptar a apresentação para pessoas com deficiência visual, como exemplo dificuldades de enxergar e diferenciar cores.

**3 -) Sintaxe básica do CSS**

A sintaxe básica do CSS é composta por um seletor (i), uma propriedade (ii) e um valor (iii). Essas três informações combinadas formam uma **regra de estilização** que será aplicada ao elemento HTML. O **seletor (i)** define qual elemento HTML será modificado. A **propriedade (ii)** define o que será alterado no elemento selecionado. O **valor (iii)** define o novo valor da propriedade. Exemplo:

```
seletor {  
  propriedade: valor;  
}
```

Por exemplo, o código abaixo define que a cor do texto de todos os elementos “<p>” será vermelho.

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

### 3.1 – Tipos de Estilização

Podemos aplicar a estilização de três formas com o CSS: *inline*, interna e externa.

#### 3.1.1 Inline

Inserir a estilização diretamente como um atributo do elemento HTML é conhecida como **estilização inline**. Esse formato não é flexível, não permite reuso e complica a legibilidade e por isso não é recomendado. No exemplo abaixo vamos aplicar vermelho em apenas um elemento “<p>” usando a estilização **inline**. Perceba que nesse exemplo a estilização será aplicada a apenas a esse único elemento e não há necessidade de informar o seletor, visto que a estilização já está no próprio elemento.

```
<p style="color:red;">Texto em Vermelho</p>
```

#### 3.1.2 Interna

Podemos também utilizar a **estilização interna**. Nesse cenário a estilização é aplicada no “head” da página e ficará disponível para todos os elementos da página. É um pouco flexível e organizada que a *inline* mas também não é a melhor opção. Pode rapidamente se tornar problemática a sua manutenção e legibilidade.

No exemplo abaixo vamos estilizar todo o texto em “<p>” para ficar vermelho.

```
<head>  
  <title>Somos todos vermelho</title>  
  <style>  
    p{  
      color: red;  
    }  
  </style>  
</head>  
<body>  
  <p>Eu sou vermelho</p>  
  <p>Eu também serei vermelho</p>  
</body>
```

Observe que usamos a tag <style> como descendente da <head>. Dentro de <style> podemos escrever as regras normalmente, usando a trinca **seletor**, **propriedade** e **valor**. Todos os elementos dentro da página estão sujeitos as regras de estilização definidas dentro de <style>.

#### 3.1.3 Externa

A estilização externa é a mais utilizada e, por regra, a melhor opção a ser utilizada. As propriedades CSS são definidas em um arquivo separado, geralmente com a extensão ".css". Para vincular esse arquivo CSS externo a uma página HTML, é usado o elemento <link> no <head> do HTML. Dentro de <link> devemos informar o caminho onde se encontra o arquivo ".css" com as regras de estilização.

No exemplo abaixo considere a existência de um arquivo chamado "styles.css" que se encontra no mesmo diretório que o arquivo "index.html".

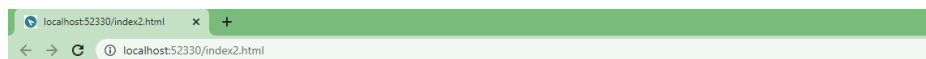
```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <h1>Exemplo de estilização externa com CSS. Eu sou azul, estou
    centralizado e minha fonte está com 32 pixels.</h1>
  <p>Eu sou vermelho, e minha fonte está com 18 pixels.</p>
  <p>Eu também sou vermelho, e minha fonte também está com 18 pixels.</p>
</body>
</html>
```

Nesse exemplo, o arquivo "styles.css" está vinculado à página HTML usando o elemento <link> e a propriedade "href". Veja que o elemento <link> é um descendente de <head>. Agora, qualquer propriedade CSS definida nesse arquivo será aplicada à página HTML. O atributo "type" não é necessário no HTML5. Abaixo temos o arquivo "styles.css":

```
p {
  color: red;
  font-size: 18px;
}

h1 {
  font-size: 32px;
  text-align: center;
  color: blue;
}
```

Para o elemento <p> estamos modificando duas propriedades: cor do texto (color) e tamanho da fonte (font-size). Para o elemento <h1> estamos modificando três propriedades: cor do texto (color), tamanho do texto (font-size) e alinhamento do texto (text-align). As propriedades CSS definidas para o seletor "p" serão aplicadas a todos os parágrafos (<p>) na página HTML, enquanto as propriedades definidas para o seletor "h1" serão aplicadas apenas ao elemento <h1> na página HTML.



**Exemplo de estilização externa com CSS. Eu sou azul, e  
com 32 pixels.**

Eu sou vermelho, e minha fonte está com 18 pixels

Eu também sou vermelho, e minha fonte também está com 18 pixels

## 3.2 – Seletores

Os seletores são usados para selecionar os elementos HTML que se deseja estilizar com CSS. Existem vários tipos de seletores que podemos usar, alguns exemplos são:

**Seletor de Elemento:** Esse é tipo de seletor que estamos usando desde o início dessa aula. Ela seleciona todos os elementos do tipo especificado. O seletor é o nome da tag HTML. Por exemplo, se desejar estilizar todos os parágrafos na página, use o seletor "p" conforme usamos nos exemplos anteriores. Basta usar o nome do elemento como seletor, e todas as regras serão aplicadas. Fizemos também com o seletor "h1" em um exemplo anterior.

**Seletor de classe:** Seleciona elementos com base na classe atribuída. Pode-se usar o seletor de classe para estilizar um grupo específico de elementos em sua página, basta que esse grupo tenha atributo informando a classe que ele pertence. Importante aqui não confundir com o conceito de classes em orientação a objetos.

Nesse contexto, a classe em CSS é uma maneira de identificar um conjunto de elementos HTML. Isso permitirá aplicar os estilos a esse conjunto específico de elementos. A classe é definida no arquivo CSS, e no arquivo HTML é necessário informar quais elementos pertencem àquela classe. No arquivo CSS a definição da classe é indicada pelo caractere ponto (.) seguido do nome da classe. Por exemplo: ".minha-classe".

O código abaixo apresenta uma classe CSS chamada ".minha-classe" que irá definir todos os textos centralizados (text-align), azul (color) e com fonte 22 px (font-size).

```
.minha-classe{
    text-align: center;
    color: blue;
    font-size: 22px;
}
```

Observe que tem um ponto (.) antes da definição da classe.

No arquivo HTML, basta adicionar essa classe como valor do atributo "class", seguido pelo nome da classe, sem o caractere ponto.

Reforçando que no HTML não se usa o caractere ponto (.), diferente do arquivo CSS.

```
<head>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <section class="minha-classe">
```

Eu sou seção e possuo o atributo class com valor "minha-classe". Isso significa que todas regras de estilização definidas em ".minha-classe" serão aplicadas em mim

```
</section>
```

```
<section class="minha-classe">
```

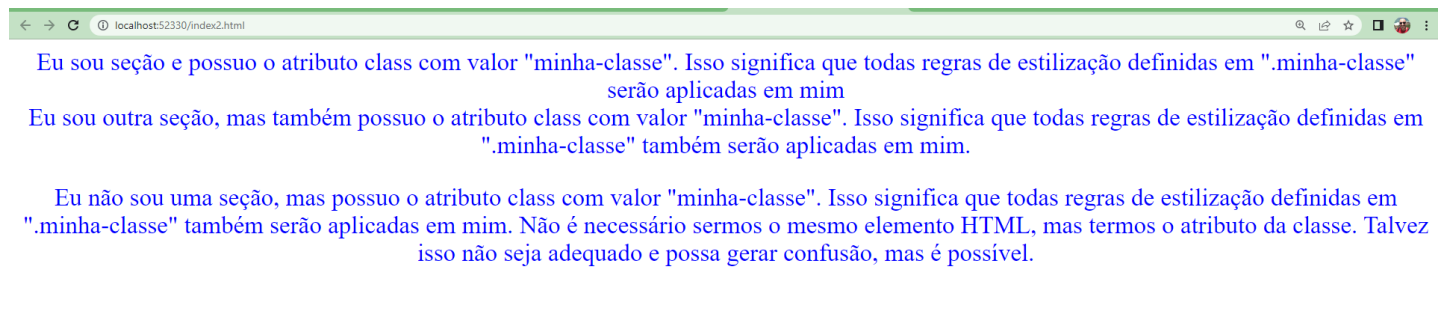
Eu sou outra seção, mas também possuo o atributo class com valor "minha-classe". Isso significa que todas regras de estilização definidas em ".minha-classe" também serão aplicadas em mim.

```
</section>
```

```
<p class="minha-classe">Eu não sou uma seção, mas possuo o atributo class com valor
"minha-classe". Isso significa que todas regras de estilização definidas em ".minha-classe"
também serão aplicadas em mim. Não é necessário sermos o mesmo elemento HTML, mas termos o
atributo da classe. Talvez isso não seja adequado e possa gerar confusão, mas é possível.</p>
</body>
```

No exemplo HTML apresentando anteriormente, todo o conteúdo das tags <section> e <p> serão afetados pelas regras definidas pela classe “minha-classe”. Isso aconteceu, pois, todos esses elementos atribuíram essa classe.

O objetivo de usar classe é facilitar o reuso de regras de estilização por diferentes elementos HTML. Normalmente esses elementos estão agrupados em <div> ou tags semânticas, de forma que a classe é atribuída nos agrupadores e não nos elementos específicos. Abaixo uma imagem do navegador apresentando o código HTML e CSS descritos acima.



**Seletor de ID:** No HTML, id (identificador) é um atributo utilizado para definir (atribuir) um valor exclusivo para determinado elemento. O CSS pode utilizar esse valor exclusivo como um identificador. Nesse caso estamos fazendo a seleção por id. A princípio, o valor desse atributo é único e não pode ser duplicado em mais elementos, isto é, não devemos ter mais de um elemento HTML com o mesmo valor de id.

Para definir um conjunto de regras para um id, colocamos o caractere “#” seguido do nome do id. No arquivo HTML atribuímos o nome sem o caractere “#”. Abaixo um exemplo de código HTML adicionando um id a um elemento <p>.

```
<body>
  <h1>Eu sou um H1 e não tenho id.</h1>
  <p id="exemplo">Eu sou um parágrafo e tenho um id. Lembrando que esse id não pode ser
duplicado. Só eu possuo esse valor</p>
</body>
```

Abaixo um exemplo do CSS usando o seletor por id. Reforçando a necessidade de colocar o caractere “#” antes. Essas regras estão modificando a cor para verde (color) e a cor de fundo para preto (background-color).

```
#exemplo {
  color: green;
  background-color: black;
}
```

Abaixo o navegador renderizando esse código CSS e HTML.



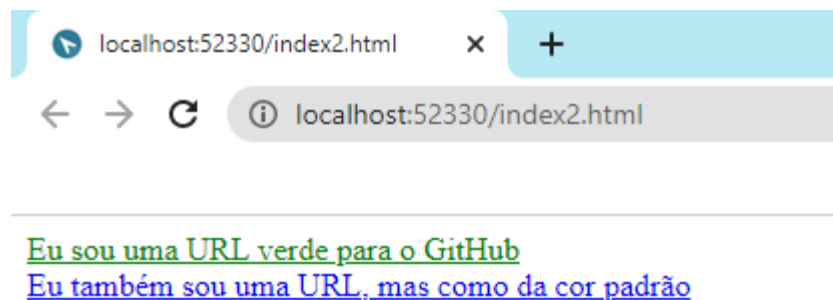
**Seletor de atributo:** Seleciona elementos com base em um atributo específico. Podemos usar o seletor de atributo para estilizar elementos que possuem um determinado atributo ou valor de atributo. No exemplo abaixo vamos criar uma regra para estilizar ancoras <a> que redirecionam para o GitHub. O link será verde ao invés de azul.

```
a[href="https://www.github.com"]{
  color:green;
}
```

Código HTML abaixo. Veja que a URL para o Google não será aplicada a regra.

```
<body>
  <div>
    <a href="https://www.github.com">Eu sou uma URL verde para o GitHub</a>
  </div>
  <div>
    <a href="https://www.google.com">Eu também sou uma URL, mas da cor padrão</a>
  </div>
</body>
```

Abaixo o navegador renderizando esse código HTML e CSS



O seletor de atributo, entre outras formas, também pode ser aplicado para elementos que contenham um atributo específico. Por exemplo, vamos estilizar parágrafos <p> que contenham um atributo id. Esses terão a cor verde.

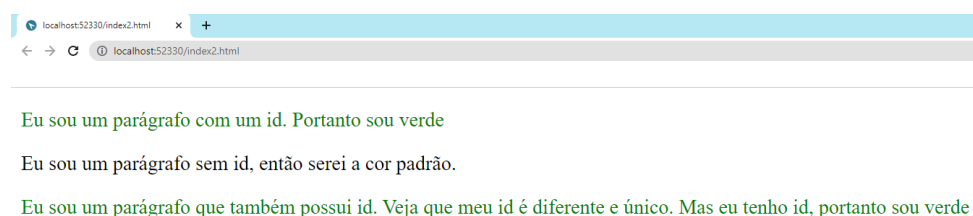
```
p[id]{
  color:green;
}
```

Perceba que tanto faz o valor do “id”, basta que ele tenha um “id”.

Código HTML abaixo:

```
<body>
  <p id="exemplo">Eu sou um parágrafo com um id. Portanto sou verde</p>
  <p>Eu sou um parágrafo sem id, então serei a cor padrão.</p>
  <p id="exemplo2">Eu sou um parágrafo que também possui id. Veja que meu id é diferente e
  único. Mas eu tenho id, portanto sou verde</p>
</body>
```

A Figura abaixo apresenta o navegador renderizando o código CSS e HTML acima.



## 4 -) Propriedades Iniciais

### 4.1 : color

A propriedade “color” permite alterar a cor do texto dos elementos HTML. Isso pode ser feito de várias formas.

**Nome da cor:** É possível atribuir um nome de uma cor, em língua inglesa, para a propriedade “color”. Exemplo para atribuir o verde.

```
p{  
    color:green;  
}
```

Apesar de parecer intuitivo, devida a quantidade de cores que as telas modernas conseguem renderizar (por volta de 16 Milhões) não é viável termos essa quantidade de nomes. E por isso temos outras alternativas para aumentarmos as possibilidades de cores.

**RGB:** As cores são compostas por componentes vermelho (Red), verde (Green) e azul (Blue). Usando sistema RGB podemos atribuir um valor, de 0-255, para cada uma dessas componentes.

Por exemplo: (255,0,0) significa R com 255, G com 0 e B com 0. Portanto o valor do vermelho (red) está no máximo e as demais zeradas. O resultado é vermelho.

```
p{  
    color: rgb(255, 0, 0)  
}
```

Podemos “brincar” com esses números para buscar a cor que mais agrada para o layout sendo elaborado.

É possível manipular também a transparência (fator/canal alfa) da cor, para isso usamos a função “rgba” e passamos mais uma componente variando de 0 até 1. Onde 0 é totalmente transparente. O exemplo abaixo adiciona opacidade de “0.2”. É quase totalmente transparente.

```
p{  
    color:rgba(255, 0, 0, 0.2)  
}
```

**Hexadecimal:** Similar ao RGB, o Hexadecimal representa cores no sistema hexadecimal. Nesse sistema temos 16 valores para representar um dígito. Variando de 0 até F. A ordem das cores é a mesma do RGB e usamos dois dígitos para cada cor. Para representar vermelho em hexadecimal fazemos #FF0000 (observe o caractere #).

```
p{  
    color: #FF0000;  
}
```

Usando o Sistema RGB ou Hexadecimal, conseguimos varrer muito possibilidades de cores e podemos usar páginas conhecidas como “color picker” e experimentar cores. Ao final será gerado um valor Hexadecimal e/ou RGB que poderemos usar na regra de estilização que estamos criando.

### 4.2 : background-color

A propriedade “background-color” permite alterar a cor de fundo do elemento HTML. Pode ser usado as mesmas formas de atribuir cor apresentado no 4.1.

#### 4.3: text-align

Descreve como o texto é alinhado no elemento pai em bloco. Isto é, o que é alinhado é o texto dentro do elemento.

Pode assumir valores como: left, right, center, justify e outros.

#### 4.4: font-weight

Controla o peso da fonte. Pode ser atribuído valores numéricos ou textuais. Algumas fontes não aceitam todos os tipos de peso, possuindo apenas normal e negrito.

```
font-weight: normal;
font-weight: bold;

/* Valor 100 é normal */
font-weight: 100;

/* Valor 700 já é negrito */
font-weight: 700;
```

#### 4.5: text-decoration-line

Adiciona uma linha decorativa no elemento textual sendo estilizado.

```
/* None pode ser usada para remover a decoração de URL */
text-decoration-line: none;
/* Por padrão a decoração da URL é underline */
text-decoration-line: underline;
text-decoration-line: overline;
text-decoration-line: line-through;
```

#### 4.6 text-decoration-style

Define o estilo da decoração adicionada com “text-decoration-line”. É uma propriedade que usamos em conjunto.

```
/* Linha sólida */
text-decoration-style: solid;
/* Linha dobrada */
text-decoration-style: double;
/* Linha pontilhada */
text-decoration-style: dotted;
/* Linha ondulada */
text-decoration-style: wavy;
```

#### 4.8 text-decoration-color

Fornece uma cor a linha adicionada com “text-decoration-line”. Utiliza-se as mesmas formas de atribuir cor já visto: nome, RGB e Hexadecimal.

#### 4.7: line-height

Define a altura de uma linha de texto, ou seja, o espaço vertical entre as linhas de um bloco de texto. Podemos especificar valores numéricos, pixels ou relativos como **em** ou porcentagem.



Exemplo: Se definirmos line-height como 1.5em ou 150%, cada linha de texto terá 1,5 vezes a altura do tamanho da fonte. Se a fonte for de 16px, a altura de linha será de 24px (16px x 1,5).