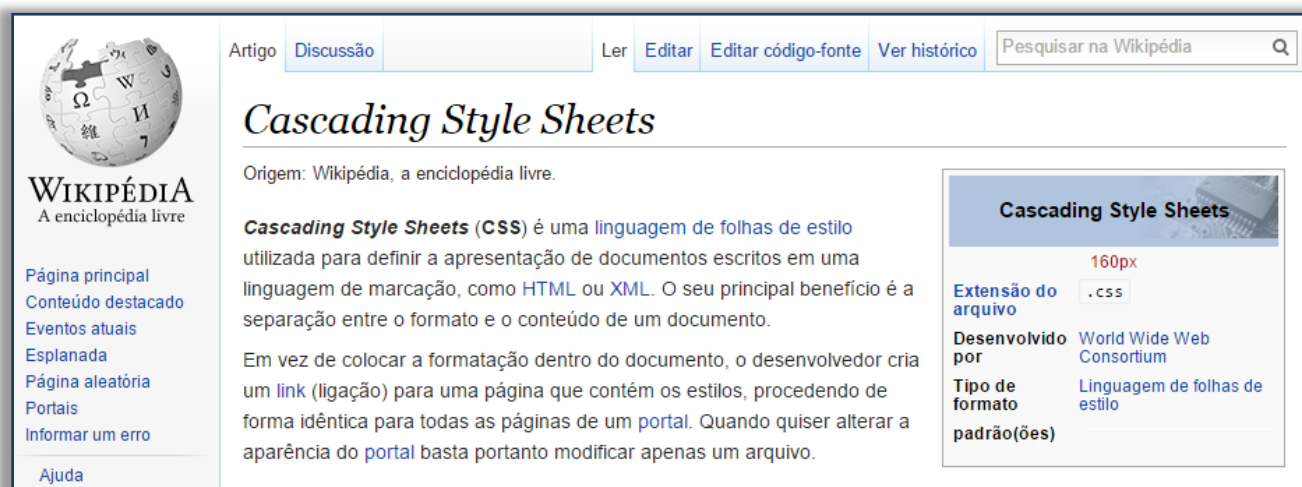


Curso de Web Developer - Básico	Aula02
Professora: Luana Fernandes	

Conteúdo desta aula:

- CSS
- Maneiras de usar o CSS
- Tipografia e alinhamento textual
- Algumas propriedades interessantes
- Id e Class
- Exercícios

CSS



SINTAXE DO CSS

A sintaxe do CSS é simples:

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

Seletor → Representa qual elemento presente no arquivo HTML você deseja selecionar para estilizar.

Propriedade → Representa qual estilização você quer fazer naquele elemento.

Valor → Representa o valor que seja usado naquela propriedade.

No código a seguir por exemplo, o seletor foi o título <h1>; a propriedade escolhida foi o color, que serve para mudar a cor de um elemento; e o valor passado para essa propriedade foi o red, que define que a cor das letras do título serão vermelhas, mas poderíamos ter usado qualquer outra cor.

```
h1 {  
    color : red;  
}
```

ATENÇÃO: É muito importante respeitar o **escopo** (bloco de chaves) de cada seletor, pois caso um seletor invada o espaço do outro, o código não vai funcionar. Veja o exemplo:

```
h1 {  
    color : red;  
    h2 {  
        color : green;  
    }  
}
```

Os seletores são a alma do CSS e você precisa dominá-los. É com os seletores que você irá escolher um determinado elemento dentre todos os outros elementos do site para formatá-lo. **E esse conhecimento será usado em vários outros momentos, como para usar Javascript, JQuery, fazer testes automatizados etc.** Boa parte da inteligência do CSS está em saber utilizar os seletores de uma maneira eficaz, escalável e inteligente.

A **sintaxe do CSS** tem uma estrutura simples composta de propriedades e valores separados pelo sinal de dois pontos (:), e cada propriedade é separada por um sinal de ponto e vírgula (;). Veja o código a seguir que é usado para definir um elemento com a cor azul e o seu fundo na cor amarela:

```
background-color : yellow;  
color : blue;
```

OBS: Embora o ponto e vírgula do último elemento seja opcional, recomendo fortemente utilizá-lo.

MANEIRAS DE USAR O CSS

1 - ATRIBUTO STYLE

A primeira delas é como um **atributo style** no próprio elemento:

```
<p style="color: blue; background-color: yellow;">  
    O conteúdo desta tag será exibido na cor azul e com o fundo amarelo!  
</p>
```

Porém **uma das grandes vantagens do CSS é manter as regras de estilo fora do HTML**. Usando o atributo **style**, não parece que fizemos isso. Justamente por isso **NÃO SE RECOMENDA ESSE TIPO DE USO NA PRÁTICA**.

2 - A TAG STYLE

A outra maneira de se utilizar o CSS é declarando suas propriedades dentro de uma **tag <style>**. Como estamos declarando as propriedades visuais de um elemento em outro lugar do nosso documento, precisamos indicar de alguma maneira a qual elemento nos referimos. Fazemos isso utilizando um **seletor CSS**, pois é uma forma de buscar certos elementos dentro da página que receberão as regras visuais que queremos.

No exemplo a seguir, usaremos o seletor que pega todas as tags <p> e altera sua cor e background:

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>Sobre a Mirror Fashion</title>  
    <style>  
      p {  
        background-color: yellow;  
        color: blue;  
      }  
    </style>  
  </head>  
  <body>  
    <p>  
      O conteúdo desta tag será exibido em azul com fundo amarelo!  
    </p>  
    <p>  
      <strong>Também</strong> será exibido em azul com fundo amarelo!  
    </p>  
  </body>  
</html>
```

Nesse exemplo estamos alterando a cor e o fundo de todos os elementos com tag <p>. Dizemos que selecionamos esses elementos pelo nome de sua tag, e aplicamos as propriedades CSS apenas neles. Porém, **ESSA FORMA DE UTILIZAÇÃO DO CSS TAMBÉM NÃO É A MAIS INDICADA.**

3 - ARQUIVO EXTERNO

A terceira maneira de declararmos os estilos do nosso documento é com um **arquivo externo**, geralmente com a **extensão .css**. Para que seja possível declarar nosso CSS em um arquivo à parte, **precisamos indicar em nosso documento HTML uma ligação entre ele e a folha de estilo.**

ALÉM DA MELHOR ORGANIZAÇÃO DO PROJETO, A FOLHA DE ESTILO EXTERNA TRAZ AINDA AS VANTAGENS DE MANTER NOSSO HTML MAIS LIMPO E DO REAPROVEITAMENTO DE UMA MESMA FOLHA DE ESTILOS PARA DIVERSOS DOCUMENTOS.

A indicação de uso de uma folha de estilos externa deve ser feita dentro da tag <head> do nosso documento HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Sobre a Mirror Fashion</title>
    <link rel="stylesheet" href="estilo.css">
  </head>
  <body>
    <p>
      O conteúdo desta tag será exibido em azul com fundo amarelo!
    </p>
    <p>
      <strong>Também</strong> será exibido em azul com fundo amarelo!
    </p>
  </body>
</html>
```

E dentro do arquivo **estilo.css** colocamos apenas o conteúdo do CSS:

```
p {
  color: blue;
  background-color: yellow;
}
```

TIPOGRÁFICAS E ALINHAMENTO TEXTUAL

A PROPRIEDADE FONTE-FAMILY

Podemos definir fontes com o uso da propriedade **font-family**. Essa propriedade recebe como valor o nome da fonte que queremos usar.

Por padrão, os navegadores exibem texto em um tipo chamado **serif**. As fontes mais conhecidas são "Times" e "Times New Roman", dependendo do sistema operacional. Elas são chamadas de **fontes serifadas** pelos pequenos ornamentos em suas terminações.

Podemos alterar a família de fontes em nosso documento para a família **sans-serif** (sem serifas), que contém, por exemplo, as fontes "Arial" e "Helvetica". Podemos também declarar que queremos utilizar uma família de fontes **monospace** como, por exemplo, a fonte "Courier".



```
h1 {  
  font-family: serif;  
}  
  
h2 {  
  font-family: sans-serif;  
}  
  
p {  
  font-family: monospace;  
}
```

É possível declararmos o nome de algumas fontes que gostaríamos de verificar se existem no computador. Existe um grupo de fontes que são consideradas "seguras" por serem bem populares. Vamos declarar essa propriedade para todo o documento por meio do seu elemento **body**:

```
body {  
  font-family: "Arial", "Helvetica", sans-serif;  
}
```

Nesse caso, o navegador verificará se a fonte "Arial" está disponível e a utilizará para renderizar os textos de todos os elementos do nosso documento. Caso a fonte "Arial" não esteja disponível, o navegador verificará a disponibilidade da próxima fonte declarada, a "Helvetica". Caso ele não a encontre, ele solicitará qualquer fonte que pertença à família **sans-serif**.

FONTES EXTERNAS

Também é possível usar fontes externas que normalmente são pagas. Existem alguns sites que oferecem fontes gratuitas, **porém é muito importante ler sobre a licença dessas fontes para ter certeza de que você pode utilizá-las no seu site**, ou se elas só podem ser usadas para fins educativos.

Um dos sites mais famosos para utilizar fontes externas é o **Google Fonts**. Nele você pode escolher uma fonte e usá-la no seu site sem problemas, mas **você não pode comercializar as fontes do Google**.

Mais informações em: <https://fonts.google.com/>

FONT-STYLE E FONT-WEIGHT

Temos outras propriedades para manipular fontes, como a propriedade **font-style**, que define o estilo da fonte que pode ser: **normal** (normal na vertical), **italic** (inclinada) e **oblique** (oblíqua). Além da propriedade **font-weight** na qual as fontes podem ser: **normal**, **bold** (negrito), ou receber um número que vai dizer o **quão negrito** você quer a letra. Teste você mesmo nos links a seguir:

- https://www.w3schools.com/cssref/pr_font_font-style.asp
- https://www.w3schools.com/cssref/pr_font_weight.asp

ALINHAMENTO E DECORAÇÃO DE TEXTO

Uma das propriedades mais simples para alterarmos as disposições dos textos é a propriedade **text-align**. O exemplo a seguir determina que todos os parágrafos da nossa página tenham o texto alinhado para a direita. Também é possível alinhá-lo ao centro (**center**) ou justificado (**justify**). O padrão é que o texto seja alinhado à esquerda (**left**).

```
p { text-align: right; }
```

É possível configurar também uma série de **espaçamentos de texto** com o CSS:

```
p {  
  line-height: 3px; /* tamanho da altura de cada linha */  
  letter-spacing: 3px; /* tamanho do espaço entre cada letra */  
  word-spacing: 5px; /* tamanho do espaço entre cada palavra */  
  text-indent: 30px; /* tamanho da margem da primeira linha do texto */  
}
```

px → Significa **pixels. As distâncias na web são normalmente medidas em pixels e não em centímetros. Falaremos disso mais à frente.*

ALGUMAS PROPRIEDADES INTERESSANTES

IMAGEM DE FUNDO

A propriedade **background-image** permite indicar um arquivo de imagem para ser exibido ao fundo do elemento. No exemplo a seguir, o navegador vai requisitar um arquivo **foto.jpg**, que deve estar na mesma pasta do arquivo CSS onde consta essa declaração.

```
h1 {  
  background-image: url(foto.jpg);  
}
```

BORDAS

Para cada borda de um elemento, podemos determinar sua cor, seu estilo de exibição e sua largura. Além disso, é possível passar os três valores juntos para a propriedade **border**. Os dois exemplos a seguir resultam no mesmo estilo:

```
body {  
  border-color: red;  
  border-style: solid;  
  border-width: 1px;  
}
```



```
body {  
  border: 1px solid red;  
}
```

COMENTÁRIOS

Conseguimos fazer também comentários no CSS usando os símbolos: **/* */**. Veja:

```
/* deixando o fundo ridículo */  
body {  
  background: gold;  
}
```

CORES NA WEB

Propriedades como **background-color**, **color**, **border-color**, entre outras aceitam uma cor como valor. Existem várias maneiras de definir cores quando utilizamos o CSS. A primeira, mais simples e ingênua, é usando o nome da cor:

```
h1 { color: red; }  
  
h2 { background: yellow; }
```

O difícil é acertar a exata variação de cor que queremos no design. Por isso, é bem incomum usarmos cores com seus nomes. O mais comum é definir a cor com base em sua composição **RGB** ou a **notação hexadecimal**.

RGB → é um sistema de cor bastante comum aos designers. Ele permite especificar até 16 milhões de cores como uma combinação de três cores base: Vermelho (Red), Verde (Green), Azul (Blue). Podemos escolher a intensidade de cada um desses três canais básicos, numa escala de 0 a 255.

Um amarelo forte, por exemplo, tem 255 de Red, 255 de Green e 0 de Blue (255, 255, 0). Se quiser a cor laranja, basta diminuir um pouco o verde (255, 200, 0). E assim por diante.

:

```
h3 {  
  color: rgb(255, 200, 0);  
}
```

Notação hexadecimal → Na notação hexadecimal (que começa com #), temos seis caracteres. Os dois primeiros indicam o canal Red, os dois seguintes, o Green, e os dois últimos, Blue.

```
h3 {  
  background: #F2EDED;  
}
```

Na base hexadecimal, os algarismos vão de zero a quinze (ao invés do zero a nove da base decimal comum). Para representar os algarismos de dez a quinze, usamos letras de A a F. Nessa sintaxe, portanto, podemos utilizar números de 0-9 e letras de A-F.

Toda vez que os caracteres presentes na composição da base se repetirem, esses podem ser simplificados. Então um número em hexadecimal **3366FF**, pode ser simplificado para **36F** por exemplo.

ID E CLASS

Apenas as *tags* não abrangem todas as possibilidades de elementos da página que podemos selecionar. Por isso, precisamos de **Ids** e **Classes**. A seguir, usamos um id chamado “assinatura” para selecionar um parágrafo e uma class chamada “titulo-principal” para selecionar um título, veja:

```
<p id="assinatura">
```

```
  _____  
  Assinatura do Cliente  
</p>
```

```
<h1 class="titulo-principal">
```

```
  Novas tendências!!  
</h1>
```


Ids e classes são os “anzóis” que precisamos para selecionar elementos específicos em um arquivo HTML. Além do CSS, outras linguagens *web*, como **Javascript**, também dependem deles. Mas quais as diferenças entre Ids e classes?

IDS SÃO ÚNICOS

- Cada elemento pode ter **apenas um id**;
- Cada página pode ter apenas **um elemento** com aquele id.

Quando se está aprendendo sobre isso, é comum ouvir que só se deve usar Ids uma vez, mas é possível usar uma classe várias vezes. Basicamente, isso entra por um ouvido e sai no outro, porque para todo mundo soa mais como uma “regrinha” do que algo realmente importante. Mas só para te avisar, você não é todo mundo, então faça o favor de usar da forma correta.

CLASSES NÃO SÃO ÚNICAS

- Você pode usar a **mesma classe** para vários elementos;
- Você pode usar **várias classes para um mesmo elemento**.

Qualquer informação de estilo que precise ser aplicada a múltiplos elementos em uma página deve ser feita com uma **classe**. Tomemos como um exemplo uma página com vários “*widgets*”:

```
1 <div class="widget"></div>
2 <div class="widget"></div>
3 <div class="widget"></div>
```

Agora, é possível usar a mesma classe “*widget*” como gancho para aplicar o mesmo conjunto de estilos para cada um deles. Porém, se preciso ter um deles maior que os outros, mas continuar com um estilo que compartilhe dos outros atributos, é possível aplicar mais de uma classe ao mesmo elemento:

```
1 <div class="widget"></div>
2 <div class="widget big"></div>
3 <div class="widget"></div>
```

CÓDIGOS DE BARRA E NÚMEROS DE SÉRIE

Talvez códigos de barra e números de série sejam uma boa analogia. Imagine um iPod, em uma loja. Na embalagem há um código de barras. Ele indica à loja que produto é esse; então, quando é escaneado, o código de barras informa exatamente que produto é aquele e quanto custa. É possível saber até sua cor e em qual lugar do armazém ele era mantido. Todos esses iPods têm exatamente o mesmo tipo de código de barras.

O iPod tem também um número de série que é absolutamente único em relação a qualquer outro iPod (ou outro dispositivo) em todo o mundo. O número de série não informa o preço. Poderia, mas para a

loja que o vende esta não seria uma maneira eficaz de armazenar e utilizar essa informação. É muito mais fácil usar um código de barras porque se, por exemplo, o preço muda, é preciso apenas alterar o preço para aquele código de barras, e não para cada número serial do sistema.

Isso é muito parecido com **ids** e **classes**. **Informações que são reusadas devem ser mantidas em uma classe e informações que são totalmente únicas devem ser mantidas em um id.**

EXERCÍCIOS

Questão 1

Desenvolva a página seguir usando HTML5 e CSS3.

O Surgimento dos Primeiros Computadores

A maioria dos grandes computadores produzidos em meados dos anos 40 até o início dos anos 50 eram não comerciais, e foram utilizados pelo governo americano, além dos fins militares para o censo, cálculos financeiros, administrativos e estatísticos, e, também, para fins científicos.

MODELO	ANO
MARK 1	1944
ENIAC	1946
EDVAC	1946
TRANSISTOR	1947

Atenção: As linhas da tabela mudam de cor ao passar o mouse por cima. Veja um exemplo:

MODELO	ANO
MARK 1	1944
ENIAC	1946
EDVAC	1946
TRANSISTOR	1947

Você deve aplicar esse efeito em todas as linhas, **exceto a linha do título da tabela.**

Dados importantes:

- Cor do título: **#ff0000**.
- Largura da tabela: **500px**.

- Cor de fundo do título da tabela: **#fd3303**.
- Cor da letra do título da tabela: **#000**.
- Cor de fundo das outras linhas da tabela: **#c48530**.
- Cor do fundo ao passar o mouse: **#999**.
- Cor da letra ao passar o mouse: **#fff**.

Questão 2

Desenvolva a página seguir usando HTML5 e CSS3.

- Cor dos títulos: **#5294a8**
- Cor do título da tabela: **#b58f83**
- Cor das linhas da tabela: **#fcedd8**
- Largura da tabela: **500px**



Pontos turísticos do Rio

O Rio é cheio de atrações! Alguns lugares são muito divulgados pela mídia e por isso as pessoas podem evitar o passeio achando que já os conhecem. Não caia nessa! Vá, sim, a todos os lugares que você puder! O Rio é uma cidade incrível com uma paisagem deslumbrante, apreciada pelo mundo inteiro. Ande, conheça, registre, faça os passeios, mesmo achando que são muito explorados, e você não se arrependerá. **Vea algumas opções:**

- A. Cristo Redentor
- B. Pão de Açúcar
- C. Orla de Copacabana
- D. Maracanã
- E. Museu do Amanhã
- F. Floresta da Tijuca

Horários de funcionamento

Ponto Turístico	Dias	Horários
Cristo Redentor	Todos	08:00 - 19:00
Pão de Açúcar	Todos	08:00 - 19:50
Orla de Copacabana	Todos	Sempre
Maracanã	Todos	08:30 - 17:00
Museu do Amanhã	Exceto às segundas	10:00 - 17:00
Floresta da Tijuca	Todos	Sempre