



CSS: Criando estilos

Introdução

O Cascading Style Sheets (CSS) é uma “folha de estilo” que define como os elementos contidos no código de uma página da internet serão exibidos. Sua maior vantagem é separar o formato do conteúdo de um documento, o que torna o HTML mais legível e mantém o documento independente do formato.

Uma folha de estilo contém definições de estilo para um ou mais documentos. Como o HTML é considerado um documento, o CSS pode ser utilizado para formatar a página HTML.

Objetivos da aula

- Descrever as principais propriedades CSS referentes a texto
- Discutir a formatação de listas para menu em CSS
- Explicar o conceito de classe em CSS

Resumo

Propriedades para texto

Com o CSS podemos definir propriedades do texto como: tamanho da fonte, cor da fonte, peso da fonte, tipo da fonte, espaçamento entre linhas e espaçamento entre letras.

p {



text-align: center;

font-family: Arial, Helvetica, sans-serif;

font-style: italic;

font-size: 14px;

text-shadow: 3px 3px 3px #ababab;

color: white;

text-decoration-line: underline;

font-weight: bold;

line-height: 90%;

background-color: green;

}

No exemplo acima, o seletor **p** vai atribuir aos parágrafos da página: alinhamento do texto ao centro, fonte Arial, tipo de fonte itálica, tamanho da fonte de 14 pixels, sombreamento ao texto, cor cinza, decoração sublinhada, negrito para peso da fonte e altura da linha de 80% do tamanho da fonte e cor de fundo verde.

A propriedade **font-family** permite que se faça uma lista de prioridades de famílias de fontes e/ou nomes genéricos de famílias a serem utilizadas. O navegador irá utilizar a primeira fonte da lista que for encontrada no computador, ou poderá fazer o download utilizando a informação contida na regra *@font-face*.

A propriedade **margin** do CSS define a área de margem nos quatro lados do elemento.



Formatando listas para menu

Um menu pode ser criado aplicando CSS aos elementos de uma lista não ordenada como a que se segue.

```
<ul>
```

```
<li><a>Item 1</a></li>
```

```
<li><a>Item 2</a></li>
```

```
<li><li>Item 3</a></li>
```

```
</ul>
```

Para transformar a lista acima em menu, removemos os marcadores utilizando a propriedade **list-style: none;** no seletor **ul**, fazemos os itens **li** "flutuarem" para a esquerda com utilizando a propriedade **float: left;** por fim, adicionamos um espaçamento entre os itens com **padding: 10px;** e removemos o sobrescrito do link com **text-decoration: none;**

```
ul {
```

```
list-style: none;
```

```
}
```

```
ul li {
```

```
float: left;
```



```
}
```

```
ul li a {
```

```
padding: 10px;
```

```
text-decoration: none;
```

```
}
```

Estilizando tabelas com pseudo-classes

Com o uso do CSS, é possível aplicar estilos a elementos por meio das pseudo-classes. Pseudo-classes são palavras-chave adicionadas aos seletores que definem um estado especial do elemento selecionado. A lista completa de pseudo-classes pode ser encontrada na documentação para desenvolvedores da Mozilla. No exemplo a seguir, uma tabela é utilizada, atribuindo a cor de fundo #e9e9e9 para as linhas pares e a cor de fundo #bdbdbd para as linhas ímpares.

Saiba mais: <https://developer.mozilla.org/pt-BR/docs/Web/CSS/Pseudo-classes>

```
tr:nth-child(even) {
```

```
background: #e9e9e9;
```

```
}
```

```
tr:nth-child(odd) {
```

```
background: #bdbdbd;
```



}

Usando caixas e atribuindo classes e ids

<div> É um elemento de divisão de conteúdo. É um contêiner genérico utilizado para agrupar elementos para fins de estilos (usando classes ou ids). Deve ser utilizada somente quando não tiver outro elemento de semântico (tal como **<section>**, **<article>**, **<header>**, **<footer>**, etc).

Classes e ids permitem ao CSS e ao Javascript selecionarem e acessarem elementos da página HTML. Classes podem ser atribuídas a vários elementos, já os ids só podem ser atribuídos a um elemento pois ele é o identificador do elemento. No exemplo a seguir, todas as divs de classe subtítulo receberão cor de fundo vermelho e cor de texto branca. E o elemento com id título receberá cor de fundo azul claro, cor de texto preta e texto alinhado ao centro.

<style>

#titulo {

background-color: lightblue;**color:** black;**text-align:** center;

}

.subtitulo {

background-color: red;



color: white;

}

</style>

<h1 id="titulo">Cidades</h1>

<h2 class="subtitulo">Londres</h2>

<h2 class="subtitulo">Paris</h2>

<h2 class="subtitulo">Tokyo</h2>

Dica quente para você não esquecer

- Existe uma **lista de fontes "seguras para a web"** (fontes que são tão comuns que são consideradas universalmente disponíveis nos computadores). Você pode encontrá-la no site da W3Schools: https://www.w3schools.com/cssref/css_websafe_fonts.asp
- Para a nomenclatura de classes e ids, é considerada boa prática usar nomes que descrevam o propósito semântico do elemento.



Conteúdo Bônus

As pseudo-classes em CSS são seletores que permitem aplicar estilos a elementos em estados específicos ou com características particulares que não podem ser selecionadas usando apenas seletores de elementos simples. Elas adicionam uma dimensão extra de seleção aos elementos, permitindo que você defina estilos com base em interações do usuário, posição na página ou outros estados.

As pseudo-classes são precedidas por um caractere de dois pontos (:) e são usadas para selecionar elementos com base em seus estados ou características que não são diretamente representadas no HTML. Algumas das pseudo-classes mais comuns incluem:

1. `:hover`: Aplica estilos quando o cursor do mouse está sobre o elemento.
2. `:active`: Aplica estilos quando o elemento está ativo (pressionado).
3. `:focus`: Aplica estilos quando o elemento está com foco, como quando é selecionado por um clique ou por meio de teclado.
4. `:nth-child(n)`: Seleciona elementos com base na posição em relação a seus irmãos.
5. `:first-child` e `:last-child`: Selecionam o primeiro e o último filho de um elemento pai.
6. `:nth-of-type(n)`: Seleciona elementos com base na posição em relação a seus irmãos do mesmo tipo.
7. `:not(selector)`: Seleciona elementos que não correspondem a um seletor especificado.

Essas pseudo-classes permitem que você crie efeitos interativos e estilizações mais dinâmicas em seu site ou aplicativo, melhorando a experiência do usuário e

proporcionando diferentes estados visuais para diferentes interações.



Por exemplo, a pseudo-classe `:hover` é frequentemente usada para criar efeitos de destaque em botões ou links quando o usuário passa o cursor sobre eles. Já a pseudo-classe `:nth-child(n)` pode ser usada para criar padrões alternados de estilo em listas ou tabelas, como listas zebradas.

Em resumo, as pseudo-classes em CSS expandem a capacidade de seleção de elementos e permitem a criação de efeitos visuais e interativos mais sofisticados em páginas da web.

Referência Bibliográfica

MDN Web Docs. [S. l.], 20 set. 2022. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>. Acesso em: 20 set. 2022.

W3SCHOOLS. [S. l.], 20 set. 2022. Disponível em: <https://www.w3schools.com/css/default.asp>. Acesso em: 20 set. 2022.

Atividade Prática 8 - CSS: Criando estilos

Título da Prática: CSS: Aplicando e Compreendendo Estilos

Objetivos: Esta atividade prática tem como objetivo fazer com que os alunos apliquem e compreendam os conceitos e técnicas de CSS aprendidos na aula “CSS: Criando estilos”.

Materiais, Métodos e Ferramentas: Editor de texto (ex: Sublime Text, Notepad++, Atom, Visual Studio Code) e navegador web.



Atividade Prática

Instruções:

1. Crie uma nova pasta em seu computador para este projeto. Nesta pasta, crie um arquivo HTML chamado “index.html” e um arquivo CSS chamado “styles.css”.

2. No arquivo “index.html”, estruture uma página HTML básica. Utilize o elemento ‘<link>’ na seção ‘<head>’ para vincular o arquivo “styles.css” à sua página HTML.

```
“html
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <link rel=“stylesheet” href=“styles.css”>
```


```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

```
”
```

3. No arquivo "index.html", adicione um cabeçalho '`<h1>`', três subtítulos '`<h2>`', uma lista não ordenada '``' com três itens '``', um parágrafo '`<p>`' e um  tabela com três linhas '`<tr>`' e duas colunas '`<td>`'. Atribua a cada elemento um id ou uma classe para que possam ser referenciados na folha de estilo CSS.

4. No arquivo "styles.css", aplique os estilos aos elementos HTML da seguinte forma:

- Para o cabeçalho '`<h1>`', use o id para definir cor de fundo, cor do texto e alinhamento do texto.

- Para os subtítulos '`<h2>`', use a classe para definir cor de fundo e cor do texto.

- Para a lista '``', use a tag para remover os marcadores de lista e flutuar os itens à esquerda. Para os links dentro da lista, defina um espaçamento e remova o sublinhado.

- Para o parágrafo '`<p>`', use a tag para definir propriedades como tamanho da fonte, cor da fonte, alinhamento do texto, tipo de fonte, sombreadimento de texto, decoração de texto, peso da fonte e altura da linha.

- Para as linhas da tabela '`<tr>`', use pseudo-classes para alternar a cor de fundo entre as linhas.

5. Depois de definir todos os estilos, salve os arquivos e abra o arquivo "index.html" em seu navegador para ver o resultado.

6. Analise como as diferentes propriedades e valores em sua folha de estilo CSS afetam a aparência de sua página HTML. Experimente com diferentes

propriedades e valores para entender melhor como cada um funciona.



Resolução:

Critérios de Avaliação:

1. Uso correto da sintaxe CSS.
2. Aplicação de diferentes tipos de seletores (id, classe, tag, pseudo-classe).
3. Uso de diferentes propriedades CSS, conforme explicado nas instruções.
4. Estrutura correta do documento HTML, com

Resolução Comentada:

1. Estrutura básica do HTML: No arquivo "index.html", criamos a estrutura básica de uma página HTML e vinculamos a folha de estilo "styles.css".

```
"html
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

```
</head>
```

```
<body>
```



<!-- Conteúdo do corpo será adicionado aqui -->

</body>

</html>

'''

2. Adição de elementos HTML: No corpo da página HTML ('<body>'), adicionamos um cabeçalho '<h1>', três subtítulos '<h2>', uma lista não ordenada '' com três itens '', um parágrafo '<p>', e uma tabela '<table>' com três linhas '<tr>' e duas colunas '<td>'. Atribuímos um id ou uma classe a cada elemento.

'''html

<body>

<h1 id="titulo">Cidades</h1>

<h2 class="subtitulo">Londres</h2>

<h2 class="subtitulo">Paris</h2>

<h2 class="subtitulo">Tokyo</h2>

<ul id="menu">

Item 1

Item 2

Item 3



<p class="texto">Este é um parágrafo de exemplo para aplicação de estilos com CSS.</p>

<table>

<tr><td>Coluna 1 - Linha 1</td><td>Coluna 2 - Linha 1</td></tr>

<tr><td>Coluna 1 - Linha 2</td><td>Coluna 2 - Linha 2</td></tr>

<tr><td>Coluna 1 - Linha 3</td><td>Coluna 2 - Linha 3</td></tr>

</table>

</body>

”

3. Aplicação de estilos no CSS: No arquivo “styles.css”, definimos os estilos para os elementos HTML.

”css

#titulo {

background-color: lightblue;

color: black;

text-align: center;

}



```
.subtitulo {
```

```
    background-color: red;
```

```
    color: white;
```

```
}
```

```
#menu {
```

```
    list-style: none;
```

```
}
```

```
#menu li {
```

```
    float: left;
```

```
    margin-right: 10px; /* adicionamos uma margem à direita para separar os itens do menu /
```

```
}
```

```
#menu li a {
```

```
padding: 10px;
```

```
text-decoration: none;
```

```
color: black; / adicionamos uma cor ao texto do link /
```



```
}
```

```
.texto {
```

```
    text-align: center;
```

```
    font-family: Arial, Helvetica, sans-serif;
```

```
    font-style: italic;
```

```
    font-size: 14px;
```

```
    text-shadow: 3px 3px 3px #ababab;
```

```
    color: white;
```

```
    text-decoration-line: underline;
```

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


```
    line-height: 90%;
```

```
    background-color: green;
```

```
}
```

```
table {
```

```
    width: 100%; / fazemos a tabela ocupar toda a largura da página /
```

border-collapse: collapse; /removemos as bordas duplas entre as células / 

}

table td {

border: 1px solid #ddd; /adicionamos uma borda às células /

padding: 8px; / adicionamos um padding para dar espaço ao texto dentro das

celulas */

}

tr:nth-child(even) {

background: #e9e9e9;

}

tr:nth-child(odd) {

background: #bdbdbd;

}

""

4. Visualização do Resultado: Ao abrir o arquivo “index.html” em um navegador, podemos ver a página HTML com os estilos aplicados.



5. Compreensão dos estilos aplicados: Cada regra CSS em nossa folha de estilo aplica um ou mais estilos a um elemento HTML em nossa página. Por exemplo, a regra ‘#titulo’ aplica estilos ao elemento HTML com o id “titulo”. Da mesma forma, a regra ‘subtitulo’ aplica estilos a todos os elementos HTML com a classe “subtitulo”. As propriedades e valores dentro de cada regra definem os estilos aplicados. Por exemplo, ‘background-color: red;’ aplica uma cor de fundo vermelha e ‘color: white;’ aplica uma cor de texto branca. As pseudo-classes ‘:nth-child(even)’ e ‘:nth-child(odd)’ são usadas para aplicar estilos a cada segunda linha (par) e cada outra linha (ímpar) de nossa tabela, respectivamente. Experimentar com diferentes propriedades e valores pode ajudar a entender melhor como cada um afeta a aparência de um elemento HTML.

Ir para exercício