



Programação Web Front-End

Aula 1 - CSS

Profa. Rosangela de Fátima Pereira Marquesone
romarquesone@utfpr.edu.br

Proposta: apresentar os conceitos iniciais sobre CSS, visando o aprendizado referente a sua forma de uso, sintaxe e propriedades referentes a estilo de texto e cores.

Objetivos: espera-se que após essa aula, você tenha habilidade para compreender os seguintes tópicos:

1. [Descobrir o que é CSS](#)
2. [Aprender a sintaxe do CSS](#)
3. [Aprender os tipos de seletores CSS](#)
4. [Aprender as formas de inclusão de CSS](#)
5. [Aprender a definir cores](#)
6. [Aprender a alterar as propriedades de texto](#)

Dicas de aprendizado:

- Execute todos os passos com atenção, compreendendo o que está sendo realizado;
- Procure não copiar código, para ter a prática de digitar o código desenvolvido;
- Pergunte quando tiver alguma dúvida;
- Mantenha um histórico dos códigos desenvolvidos, seja no github ou em algum outro meio de armazenamento (e-mails, google drive, etc.);
- Tenha curiosidade e explore os recursos apresentados.

Tópicos anteriores:

- Compreender o que é HTML
- Compreender o que são tags HTML básicas
- Criar um arquivo .html no Visual Studio (VS) Code
- Abrir o arquivo .html em um navegador
- Visualizar o código-fonte de uma página em um navegador
- Inspeccionar a página em um navegador
- Utilizar o Live Server no VS Code
- Aprender a utilizar tags semânticas
- Aprender a inserir links
- Aprender a inserir listas
- Aprender a criar uma página com seu Curriculum Vitae (CV) (atividade prática)
- Aprender a inserir figuras
- Aprender a utilizar a tag semântica <figure>
- Inserir figuras em seu Curriculum Vitae (CV) (atividade prática)
- Aprender a criar formulários

- Criar um formulário (atividade prática)

Passo 1 - Descobrir o que é CSS

Quando iniciamos o processo de aprendizado no contexto de Programação Web Front-end, HTML é a primeira linguagem da qual temos contato, uma vez que é via HTML que inserimos o conteúdo e definimos a estrutura de uma página Web. O passo seguinte nesse processo de aprendizado é atuar com CSS, para que possamos encontrar meios para adicionar estilo à página Web desenvolvida.

Cascading Style Sheets (CSS), ou ainda Folhas de Estilo em Cascata, é um mecanismo usado para estilizar e tornar as páginas Web personalizadas. Ou seja, a partir do CSS, você poderá:

- Descrever como os documentos devem ser apresentados, permitindo realizar ações como alterar a fonte, a cor e o tamanho e o espaçamento do seu conteúdo; adicionar animações e adicionar efeitos decorativos;
- Descrever regras de como você deseja estilizar o conteúdo de uma página Web;
- Propor uma maneira que todos podem seguir para fazer escolhas de estilo em um determinado projeto.

Dentro desse contexto, podemos destacar as seguintes vantagens em utilizar CSS:

- O uso de CSS separa o conteúdo da apresentação;
- Criadores de conteúdo não precisam se preocupar com a apresentação;
- Possibilidade de reuso de dados;
- Possibilidade de customizar o estilo de acordo com as preferências e necessidades do usuário;
- Uma única folha de estilo pode definir a consistência de uma coleção de documentos.

Para saber mais!

A W3C é a fundação responsável pela padronização do CSS. Sua primeira versão aprovada pela W2C foi criada em 1996. Atualmente, a terceira e mais nova versão é a CSS3, que foi iniciada em 2012 e vem recebendo atualizações constantes. Você pode conferir os detalhes de cada versão diretamente pelo link da [W3C](#). Caso se interesse, você pode também visualizar um dos primeiros artigos sobre CSS, publicado em 1996, por esse [link](#).

Passo 2 - Aprender a sintaxe do CSS

Agora que você já compreendeu o que é e para que serve CSS, podemos entender um pouco mais sobre sua sintaxe.

Basicamente, o código CSS é definido por um conjunto de **seletores**.

Um seletor representa uma estrutura usada como uma condição, que determina quais e de que forma elementos HTML serão estilizados. A sintaxe básica de um seletor é:

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

No qual:

- **seletor**: representa uma estrutura, utilizada como uma condição para determinar quais elementos de um grupo serão formatados. Exemplo: H1, p, body, ...
- **propriedade**: representa a característica que você deseja modificar no elemento. Exemplo: fonte, cor, tamanho, ...
- **valor**: representa valor referente a esta característica. Exemplo: Arial, red, 12 px, ...

Para contextualizar melhor, veja alguns exemplos de seletores:

```
body {  
  color: red;  
}  
  
p {  
  font-family: arial;  
  text-align: center;  
}  
  
h1 {  
  background: blue;  
}
```

O valor associado a uma propriedade (exemplo: a cor vermelha ao atributo color), é chamado de **declaração CSS**. Veja que as declarações CSS são colocadas em blocos de declaração, que por sua vez, são pareados com os seletores, produzindo conjuntos de **regras CSS**. No exemplo anterior, o conjunto de regras CSS do seletor p possui duas declarações.

Lembra do objeto DOM, que vimos em HTML? Em CSS, para aplicar o estilo em uma página Web, o navegador é responsável por converter as regras CSS em um mapa de estilos CSS Object Model (CSSOM). A partir desse mapa, ele passa por cada regra definida no CSS, criando uma árvore de nós, com base nos seletores CSS que foram especificados. Ou seja, ocorre um processo similar ao objeto DOM, para renderizar o conteúdo HTML.

Você também pode estar se perguntando o que são as propriedades de um seletor. Definir quais propriedades você deverá utilizar e quais valores atribuir a elas é o elemento central no desenvolvimento de um código CSS.

No decorrer desse módulo da disciplina veremos mais sobre as propriedades, mas, por enquanto, saiba quais são algumas das mais comumente utilizadas:

- **background**: propriedade abreviada para definir os valores da cor de fundo em um elemento na folha de estilo.
- **background-image**: define uma imagem de fundo.
- **color**: define o valor da cor de um elemento com seu conteúdo em texto.
- **font-size**: define o tamanho da fonte.
- **font-family**: define lista de prioridades de famílias de fontes e/ou nomes genéricos de famílias de fontes.
- **font-style**: define propriedades de estilos (normal, italic ou oblique).
- **text-align**: controla o posicionamento horizontal do conteúdo de um elemento (left, right, center, justify).
- **text-decoration**: define efeito decorativo em um texto. Exemplos: underline (sublinhado); line-through (linha cortando o texto); e blink (efeito piscante).
- **text-transform**: controla os efeitos de capitalização do texto. Exemplos: capitalize, uppercase, lowercase.
- **width**: determina a largura da área de conteúdo de um elemento.

Esses são somente alguns exemplos das propriedades disponíveis quando você for estilizar sua página. Você pode encontrar a documentação contendo uma lista completa das propriedades por meio deste [link](#).

Também é importante saber que, diferente do arquivo HTML, para inserir um comentário em um arquivo CSS, utiliza-se o /* e */, conforme exemplo a seguir:

```
/*inserindo um comentário em arquivo CSS*/  
h1 {  
    background: blue; /*inserindo outro comentário*/  
}
```

Passo 3 - Aprender os tipos de seletores CSS

Como foi visto no passo anterior, um seletor é responsável por definir o estilo de um ou mais elementos HTML. Para isso, os seletores mais utilizados são os seguintes:

- Seletor por elemento HTML;
- Seletor por ID;
- Seletor por classe;
- Seletor por atributo;
- Seletor por pseudo-classe;
- Seletor universal.


Vamos ver um exemplo de cada um deles?

Seletor por elemento HTML

Aplica o estilo aos elementos pelo nome do elemento HTML (exemplo: p, body, div, H1, img). Ele seleciona e aplica a regra CSS a todos os elementos do tipo fornecido em um documento.

Sintaxe: *elemento { propriedades de estilo }*

No exemplo a seguir, será alterada a cor da fonte (propriedade *color*) para vermelho, de todos os parágrafos (tag <p>). Também será inserida a cor amarelo como cor de fundo (propriedade *background*) de toda a página (tag <body>).

CSS	HTML	Resultado
<pre>p { color: red; } body { background: yellow; }</pre>	<pre><body> <p> Aula de CSS </p> </body></pre>	

Seletor por ID

Uma das coisas que precisamos recapitular sobre HTML é que os elementos podem ter atributos. Um desses atributos refere-se ao id, que deve ser único para cada elemento. Ou seja, esse atributo especifica um **ID exclusivo** para um elemento HTML, portanto, seu valor deve ser exclusivo no documento.

Nesse contexto, utilizamos o seletor por ID em casos onde há apenas um elemento HTML com um determinado valor no atributo ID e desejamos aplicar a regra de estilo somente a esse elemento.

Sintaxe: *#nome_id { propriedades de estilo }*

Veja que o seletor deve ser precedido por uma hashtag (#) e seu nome deve ser exatamente o mesmo que estiver descrito no atributo ID de um elemento HTML, conforme o exemplo a seguir.

CSS	HTML	Resultado
<pre>#t1 { color: red; } #t2 { color: orange; font-style:italic; }</pre>	<pre><h1 id="t1">Aula de CSS</h1> <h2 id="t2">tipos de seletores</h2></pre>	<p>Aula de CSS</p> <p><i>tipos de seletores</i></p>

Seletor por classe

Além do atributo ID, um elemento HTML também pode possuir um atributo chamado *class*, que corresponde a uma classe do elemento. Diferente do ID, um valor de atributo class pode ser aplicado a mais elementos.

Dessa forma, o seletor por classe aplica as regras CSS aos elementos HTML que possuam o atributo *class* especificado.

Sintaxe: *.nome_classe { propriedades de estilo }*

Perceba que nesse seletor, o nome da classe deve ser precedido pelo ponto, conforme exemplo a seguir.

CSS	HTML	Resultado
<pre>.destaque { color: red; font-size:200%; } .avisos { background: yellow; }</pre>	<pre><p class="destaque">Aula de CSS</h1> <p class="avisos">Dia de avaliação</h2> <p class="avisos">Revise o conteúdo</h2></pre>	<p>Aula de CSS</p> <p>Dia de avaliação</p> <p>Revise o conteúdo</p>

Veja que os dois últimos parágrafos possuem o mesmo valor do atributo classe ("avisos"). Dessa forma, a regra CSS (.avisos) com seletor por classe será aplicada a ambos.

Seletor por atributo

Vimos que os elementos HTML possuem diferentes tipos de atributos. Por exemplo, o elemento `<a>` possui o atributo href e o elemento `` possui o atributo src. Em casos onde desejamos aplicar a regra CSS somente quando um atributo de um elemento HTML tiver um valor específico, utilizamos esse tipo de seletor.

Sintaxe: *elemento[atributo=valor] { propriedades de estilo }*

Veja um exemplo a seguir, no qual é aplicado uma regra CSS que torna a cor de fundo amarelo somente ao elemento <a> cujo atributo href tenha o valor “https://www.css.com”.

CSS	HTML	Resultado
<pre>a { color: green; } a[href="http://www.css.com"] { background-color: yellow; }</pre>	<pre> Link da aula Link especial </pre>	<ul style="list-style-type: none">• Link da aula• Link especial

Seletor por pseudo-classe

Uma pseudo-classe é usada para definir um estado especial de um elemento. Um exemplo desse estado pode ser identificado em links (<a>), que podem ter um estado quando o link é visitado ou quando se passa o mouse sobre ele.

Dessa forma, esse seletor é utilizado indicando o nome do elemento, seguido de dois pontos e uma palavra-chave que especifica um estado especial do elemento HTML selecionado.

Sintaxe: *seletor:pseudo-classe { propriedade: valor }*

Veja um exemplo a seguir, no qual, ao clicar com o mouse sobre o link (estado *hover*), a cor do texto é alterada para vermelho.

CSS	HTML	Resultado
<pre>a { color: green; } a:hover { color: red; }</pre>	<pre> Link da aula Link especial </pre>	<ul style="list-style-type: none">• Link da aula• Link especial

Seletor universal

Por fim, utiliza-se o seletor universal quando se deseja aplicar uma regra CSS a todos os elementos de uma página. Para isso, o asterisco (*) é utilizado como nome do seletor.

Sintaxe: ** { propriedades de estilo }*

Veja um exemplo a seguir, no qual é aplicada a cor de fonte verde para todos os elementos da página.

CSS	HTML	Resultado
<pre>* { color: green; }</pre>	<pre><h1>Aula de CSS</h1> <p>Seletor universal</p></pre>	<p>Aula de CSS</p> <p>Seletor universal</p>

Além de saber como utilizar os diferentes tipos de seletores, é importante saber que também é possível realizar o agrupamento de seletores. Ou seja, ao invés de indicar a regra CSS em cada seletor, você pode agrupá-los, tornando seu código mais limpo. Veja alguns exemplos:

Seletor	Exemplo	Descrição
.class1.class2	<pre>.nome1.nome2{ color:red; }</pre>	Seleciona todos os elementos com nome1 e nome2 definidos em seu atributo de classe.
elemento, elemento	<pre>div, p { background: green; }</pre>	Seleciona todos os elementos div e todos os elementos p.
elemento elemento	<pre>div p { background: green; }</pre>	Seleciona todos os elementos p que estão em um elemento div.

Passo 4 - Aprender as formas de inclusão de CSS

Além dos tipos de seletores, também é importante que você saiba que existem 3 formas de inserir o código CSS em um documento:

- Folha de estilo externa
- Folha de estilo interna
- Estilo inline

Folha de estilo externa

Considerado atualmente o caso mais comum, refere-se a quando você define seu CSS escrito em um arquivo separado, com extensão .css, no qual você o refere dentro de um elemento <link> do documento HTML, conforme exemplo a seguir:

style.css	index.html
<pre>h1 { color: blue; background: yellow; } p { color: red; }</pre>	<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8" /> <title>Exemplo</title> <link rel="stylesheet" href="css/style.css" /> </head> <body> <h1>Aula de CSS!</h1> <p>Criando CSS externo</p> </body> </html></pre>

Veja que o atributo href do elemento <link> precisa fazer referência a um arquivo do seu projeto. Nesse cenário, o arquivo estava dentro da pasta css.

Folha de estilo interna

Nesse caso, o código fica no mesmo arquivo que o documento HTML. Para isso, você deve colocar o código CSS dentro de elemento <style>, localizado no elemento <head> do documento HTML, conforme exemplo a seguir, destacado em vermelho:

index.html
<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8" /> <title>Exemplo</title> <style> h1 { color: blue; background: yellow; } </style> </head> </html></pre>

```
p {
  color: red;
}
</style>
</head>
<body>
  <h1>Aula de CSS!</h1>
  <p>Criando CSS externo</p>
</body>
</html>
```

Embora essa opção seja útil em alguns cenários, como quando você não tem acesso a edição de arquivos .css, a melhor opção ainda é a de uso de CSS externo, dado que o CSS pode ser atualizado em vários arquivos HTML sempre que alguma alteração for necessária.

Estilo inline

Por fim, uma outra opção, menos indicada, é a de CSS inline, sendo essas declarações CSS que afetam apenas um determinado elemento, de forma que o conteúdo CSS ocorre por um atributo *style*, conforme o exemplo a seguir.

index.html
<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8" /> <title>Exemplo</title> </head> <body> <h1 style="color: blue; background: yellow;">Aula de CSS!</h1> <p style="color:red;">Criando CSS externo</p> </body> </html></pre>

Caso você pesquise por livros ou sites referentes à CSS, verá alguns avisos de alerta do tipo: “**Não utilize essa opção, a menos que seja estritamente necessária!**”. Das 3 opções apresentadas, essa é a que mais dificulta a manutenção do código, dado que as regras de estilo estão espalhadas ao longo do documento HTML, dificultando sua compreensão.

Bem, mas será que posso ter em um arquivo HTML a inclusão de CSS externo, interno e inline? A resposta é sim. Por esse motivo, você precisa saber também que há uma ordem de prioridade na aplicação do estilo, que ocorre em cascata. Além disso, caso diferentes estilos forem definidos para um mesmo elemento, utilizando o mesmo seletor, prevalece o de maior prioridade, conforme a ordem a seguir:

1. Estilo inline
2. Folha de estilo interna
3. Folha de estilo externa
4. Definição do navegador (*browser*)

Além disso, atenha-se as outras seguintes regras também:

- Caso haja a declaração tanto de folha de estilo externa quanto interna, a prioridade dependerá da ordem de declaração.
- Caso um seletor seja definido em mais de uma folha de estilo externa, a última referenciada terá a precedência.
- O navegador também selecionará a última opção de estilo, caso o seletor seja definido mais de uma vez.
- Para mudar essa ordem, utiliza-se a regra *!important* logo após o valor de uma propriedade, para sobrescrever as demais regras para o elemento.

Para saber mais!

Todos os fatores apresentados até agora referem-se a 3 importantes características do CSS: cascata, especificidade e herança. Você pode ler mais sobre cada um deles por este [link](#).

Passo 5 - Aprender a definir cores

Um dos passos importantes no CSS é aprender a definir e utilizar cores.

Para definir o valor de uma propriedade referente a uma cor, é possível utilizar as seguintes opções:

- **Palavra-chave (em inglês):** nome predefinido da cor, como black, white, blue, violet, etc..
- **Escala RGB (vermelho, verde, azul):** define os valores de 0 a 255, para R (vermelho), G (verde) e B (azul). Exemplo: rgb(0,0,0) para preto e rgb(255,255,255) para branco.
- **Valores hexadecimais:** valores para R, G e B definidos com números hexadecimais (de 00 a FF, ou de 0 a 255). Exemplo: #000000 para preto e #FFFFFF para branco.
- **Valores de saturação:** permite definir as cores com uso de parâmetros como saturação e luminosidade.

Além disso, você também consegue definir fatores como a transparência da cor e sua luminosidade, conforme o exemplo a seguir.

```
body {  
  background-color: yellow; /* Cor amarela */  
  
  background-color: #409767 /*Pares hexadecimais RGB*/  
  
  background-color: rgb(255, 0, 0); /* Cor vermelha */  
  
  background-color: rgba(255, 0, 0, 0.7) /* Cor vermelha, com 70% de transparência */  
  
  background-color: hsl(15, 80%, 50%) /* (matiz (0-360), saturação, luminosidade) */  
  
  background-color: hsla(15, 80%, 50%, 0.5) /*(matiz(0-360), saturação, luminosidade, transparência) */  
}
```

Praticando:

- Faça o download do arquivo compactado “projetos-web-css.zip”, disponível no moodle da disciplina.
- Descompacte o arquivo e abra-o no VS Code.
- Abra o arquivo exemplo1.html no navegador utilizando Live Server e visualize o estilo da página, conforme a figura a seguir:

Objetivos de Desenvolvimento Sustentável

[Pacto Global](#) | [Link da ONU](#)

Objetivos de Desenvolvimento Sustentável (ODS)



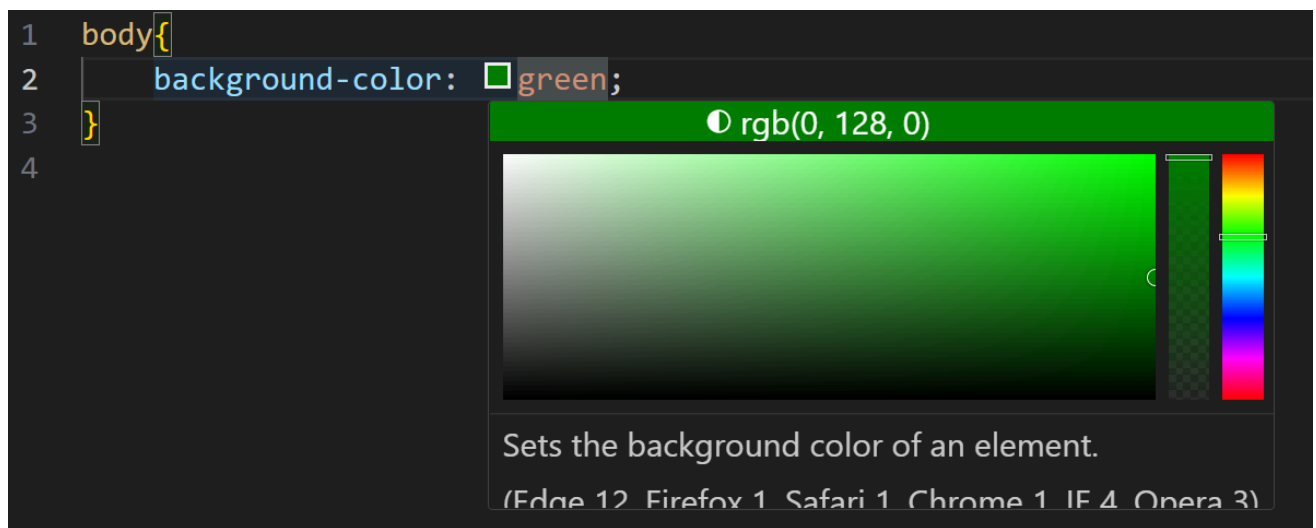
Perceba que o fundo da página está amarelo. Essa cor foi definida no arquivo `style.css`, que está na pasta `css` do projeto, conforme o código a seguir:

css/style.css
<pre>body{ background-color: yellow; }</pre>

Altere o arquivo, incluindo novas cores para os elementos `header` e `footer`, conforme exemplo a seguir:

css/style.css
<pre>body{ background-color: green; } header{ /*insira uma cor de fundo aqui*/ } footer{ /*insira uma cor de fundo aqui*/ }</pre>

Veja que o VS Code auxilia na seleção de cores, apresentando uma opção de paleta para você, conforme a figura a seguir:



Ao clicar na descrição da cor, é apresentada outras opções de formato, como hexadecimal, hsl e hwb.

Bem, é importante que você saiba que a propriedade background-color não é a única que podemos utilizar para definir uma cor. Podemos, por exemplo, usar a propriedade color para definir a cor de um elemento textual, ou para definir a cor de uma borda.

Para exemplificar, inclua o código a seguir no seu arquivo css:

css/style.css
<pre>h1{ color: blue; }</pre>

O resultado no navegador deve mostrar o texto dos elementos h1 em azul, similar à figura a seguir:

Objetivos de Desenvolvimento Sustentável

[Pacto Global](#) | [Link da ONU](#)

Objetivos de Desenvolvimento Sustentável (ODS)

The infographic displays the 17 Sustainable Development Goals (ODS) in a grid format. Each goal is represented by a numbered icon and a brief description in Portuguese:

- 1. ERRADICAÇÃO DA POBREZA
- 2. FOME ZERO E AGRICULTURA SUSTENTÁVEL
- 3. SAÚDE E BEM-ESTAR
- 4. EDUCAÇÃO DE QUALIDADE
- 5. IGUALDADE DE GÊNERO
- 6. ÁGUA POTÁVEL E SANEAMENTO
- 7. ENERGIA LIMPA E ACESSÍVEL
- 8. TRABALHO DECENTE E CRESCIMENTO ECONÔMICO
- 9. INDÚSTRIA, INOVAÇÃO E INFRAESTRUTURA
- 10. REDUÇÃO DAS DESIGUALDADES
- 11. CIDADES E COMUNIDADES SUSTENTÁVEIS
- 12. CONSUMO E PRODUÇÃO RESPONSÁVEIS
- 13. AÇÃO CONTRA A MUDANÇA GLOBAL DO CLIMA
- 14. VIDA NA ÁGUA
- 15. VIDA TERRESTRE
- 16. PAZ, JUSTIÇA E INSTITUIÇÕES EFICAZES
- 17. PARCERIAS EM DESENVOLVIMENTO

Os Objetivos de Desenvolvimento Sustentável são um apelo global à ação para acabar com a pobreza, proteger o meio ambiente e o clima e gar

Como as Nações Unidas apoiam os Objetivos de Desenvolvimento Sustentável no Brasil

Praticando: altere o arquivo style.css, incluindo uma nova cor para os elementos <p> e <h2>.

Praticando: verifique se seu código CSS está adequado por meio do [validador de CSS](#) da W3C.

Para saber mais!

É importante lembrarmos também que, além da questão de aparência, muitas pessoas possuem dificuldade em distinguir cores, portanto, por questões de **acessibilidade**, é altamente recomendado a escolha de cores que tenham um bom contraste. Você pode encontrar mais informações sobre a escolha de cores nos links a seguir:

- [Contrast Checker](#): ferramenta para analisar o contraste das cores.
- [Teste de contraste de cores](#): tutorial para fazer o teste de contraste por meio das ferramentas do desenvolvedor que o navegador oferece.
- [Color Picker](#): ferramenta de seleção de cores da W3School.
- [CSS Gradient](#): oferece a opção de gerar imagens com a definição de cores gradientes.

Caso tenha outras ferramentas e materiais interessantes que queria compartilhar com a turma, crie uma postagem no fórum “Dicas de conteúdo”, disponível no moodle da disciplina.

Curiosidade!

Em busca de demonstrar o poder do CSS, foi criado em 2003 o projeto [CSS zen garden](#), referente a uma galeria colaborativa com foco em mostrar as diferentes formas que uma página HTML pode ter, a partir do CSS. Designers colaboraram apresentando diferentes versões de uma mesma página, com o design realizado apenas através de CSS e imagens/fontes.

Vale a pena conferir alguns exemplos para se inspirar!

Passo 6 - Aprender a alterar as propriedades de texto

Uma outra habilidade que você deve adquirir ao utilizar CSS é a de estilizar texto. Isso inclui saber trabalhar com cores, tipos de fontes, tamanhos e efeitos decorativos do seu texto. É essa habilidade que você irá adquirir nessa etapa. Vamos lá?

Para isso, os seguintes tópicos serão abordados:

- Fontes
- Decoração
- Alinhamento

Fontes

Após a definição de cores do texto, a fonte é uma propriedade comumente utilizada para acrescentar estilo à página Web.

Uma das propriedades utilizadas para definir a fonte é a **font-family**. Ela permite especificar uma lista de uma ou mais nomes de famílias de fontes (como "Times New Roman", "Courier", "Arial", etc.) e/ou nomes de famílias genéricas (como "serif", "sans-serif" e "cursive") para o elemento. Você pode conferir uma lista de fontes e seu formato por este [link](#).

Esse conjunto é definido para que, caso o navegador não suportar a primeira fonte, ele tentará a próxima fonte. Por isso, quando for declarar essa propriedade, separe cada nome por uma vírgula e sempre declare por último na lista o nome de fonte genérico, para ser utilizado somente em última opção.

Veja alguns exemplos a seguir:

```
font-family: Times, "Times New Roman", Georgia, serif;  
font-family: Verdana, Arial, Helvetica, sans-serif;  
font-family: "Lucida Console", Courier, monospace;
```

Veja a seguir outras opções de formatação de fonte disponíveis:

- **font-size**: estabelece o tamanho da fonte.
 - Exemplos:
 - font-size: small; /*nomes possíveis: xx-small, x-small, small, medium, large, x-large, xx-large*/
 - font-size: 12px; /*tamanho definido em pixels.*/
 - font-size: 0.8em; /*1em equivale a 100%, ou seja, ao tamanho padrão de fonte.*/
 - font-size: 80%; /*porcentagem relativa ao tamanho de fonte padrão do navegador do usuário*/

Observação: por questões de acessibilidade, geralmente é melhor usar valores relativos ao tamanho de fonte padrão do usuário.

- **font-style**: usada para definir o estilo da fonte, podendo receber o valor normal, italic e oblique.

- Exemplos:
 - `font-style: italic;`
 - `font-style: normal;`
 - `font-style: oblique;`
- **font-weight**: utilizado para definir o peso ou a intensidade da fonte (exemplo: deixar a fonte em negrito)
 - Exemplo:
 - `font-weight: bold;`

Visualize o resultado no arquivo `exemplofonte.html`, abrindo o arquivo no navegador via Live Server, conforme a imagem a seguir:



Aprendendo CSS

Aprendendo CSS

Aprendendo CSS

Praticando: inclua mais dois parágrafos, formatando as fontes de acordo com sua preferência.

Para saber mais!

Além das fontes apresentadas, você também pode incluir fontes externas em seu código CSS, como, por exemplo, as fontes do [Google Fonts](#). Para acessar o modo de inclusão, selecione uma fonte e acesse a opção “*Get embed code*”

Após isso, utilize a tag `link` para indicar a referência da fonte. Exemplo:

```
<link
href="https://fonts.googleapis.com/css2?family=Bubblegum+Sans&family=Markazi+Text:wght@400..700&
display=swap" rel="stylesheet">
```

Use também o seletor de classe indicado pelo site do Google Fonts para aplicar em seu texto, conforme exemplo a seguir:

```
.bubblegum-sans-regular {
font-family: "Bubblegum Sans", sans-serif;
font-weight: 400;
font-style: normal;
}
```

Esse nome deverá ser aplicado a uma tag de texto de seu código HTML, conforme exemplo a seguir:

```
<h1 class="bubblegum-sans-regular">Testando a fonte do Google</h1>
```

O resultado será similar ao exemplo a seguir:

Testando a fonte do Google

Decoração

Além de controlar aspectos da fonte, o CSS também permite adicionar efeitos de decoração no texto (exemplo: sublinhado, sobrescrito). Isso é feito a partir da propriedade ***text-decoration***, que refere-se a uma propriedade abreviada das propriedades *text-decoration-line* (obrigatória), *text-decoration-color* e *text-decoration-style*.

Confira alguns exemplos de decoração a seguir:

- `text-decoration: underline;`
- `text-decoration: overline blue;`
- `text-decoration: underline overline;`
- `text-decoration: overline;`
- `text-decoration: line-through;`
- `text-decoration: underline dotted;`
- `text-decoration: underline dotted red;`
- `text-decoration: red wavy underline;`
- `text-decoration-style: underline solid blue;`
- `text-decoration-style: underline double;`
- `text-decoration-style: underline dashed;`
- `text-decoration-style: underline wavy;`

Visualize o resultado no arquivo `exemplotextdecoration.html`, abrindo o arquivo no navegador via Live Server, conforme a imagem a seguir:

Aprendendo CSS

Aprendendo CSS

Aprendendo CSS

~~Aprendendo CSS~~

Aprendendo CSS

Aprendendo CSS

Aprendendo CSS

Aprendendo CSS

Aprendendo CSS

Aprendendo CSS

Aprendendo CSS

Além desta propriedade, podemos adicionar também uma sombra ao texto, a partir da propriedade ***text-shadow***. Essa propriedade requer 3 medidas, podendo também ser adicionada uma cor:

- Medida 1: deslocamento da sombra para a direita (valor positivo) ou para a esquerda (valor negativo);
- Medida 2: deslocamento da sombra para baixo (valor positivo) ou para cima (valor negativo);
- Medida 3: raio para efeito *blur* na sombra;
- Cor: define a cor da sombra.

Essa propriedade aceita uma lista de sombras. Para isso, as informações de cada sombra devem ser separadas por vírgula.

Veja alguns exemplos:

- `text-shadow: 2px 2px 8px red;`
- `text-shadow: 0 0 3px red, 2px 2px 8px yellow;`
- `text-shadow: 10px 10px 3px green;`

Visualize o resultado no arquivo `exemploshadow.html`, abrindo o arquivo no navegador via Live Server, conforme a figura a seguir:

Aprendendo CSS

Aprendendo CSS

Aprendendo CSS

Praticando: inclua mais dois parágrafos, criando sombras de acordo com sua preferência.

Alinhamento

Por fim, podemos identificar que é possível também definir como o texto ficará horizontalmente alinhado. Isso é realizado pela propriedade **text-align**, que pode receber os seguintes valores:

- Alinhado à esquerda: text-align: left;
- Alinhado à direita: text-align: right;
- Centralizado: text-align: center;
- Justificado: text-align: justify;
- Inicial: configura a propriedade padrão. text-align: initial;
- Herdado: herda a propriedade do elemento pai. text-align: inherit;

Outra propriedade útil é a **text-indent**, que especifica o recuo da primeira linha em um bloco de texto. Veja alguns exemplos a seguir:

- text-indent: 20px;
- text-indent: 30%;

Praticando: inclua o código a seguir em seu arquivo style.css, para visualizar algumas dessas propriedades aplicadas ao arquivo exemplo1.html.

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

h1{
  color: blue;
  text-decoration: underline dotted;
}
```

```

footer{
  text-align: center;
}

a{
  font-size: 150%;
  font-family: Arial, Helvetica, sans-serif;
}

p{
  font-style: italic;
  font-size: 120%;
  font-family: Arial, Helvetica, sans-serif;
}

.desc{
  color: white;
  text-shadow: 0 0 4px black;
}

#sigla{
  font-weight: bold;
  text-decoration: underline overline blue;
}

#ods{
  color: rgb(16, 37, 197);
  text-decoration-style: double;
}

```

O resultado deve ser similar à figura a seguir:



Praticando: altere o conteúdo do CSS utilizado no arquivo exemplo1.html, aplicando os conceitos vistos neste tutorial. Tente apresentar um layout diferente do apresentado, demonstrando o poder do CSS.

Considerações finais

Caso tenha chegado até aqui, você conseguiu completar o conteúdo do primeiro tutorial sobre CSS. A partir desses recursos, você passa a ser capaz de tornar suas páginas mais bonitas e agradáveis. Mas há muito mais para ser aprendido sobre CSS ainda. Continue explorando as opções disponíveis, utilizando sua criatividade.

Bom estudo!