



Aula 08

# Outras Propriedades do CSS



**Certified  
Developer**  
The Ultimate Tech Degree

**DigitalHouse** >  
Coding School



# Temas

**1**

**Cores**

**2**

**Backgrounds**

**3**

**Fontes**

**1** | **Cores**



**Cores  
predefinidas**

**Mais cores**



**Objetivo**

**Aplicação**

**Revisão e  
Conclusão**

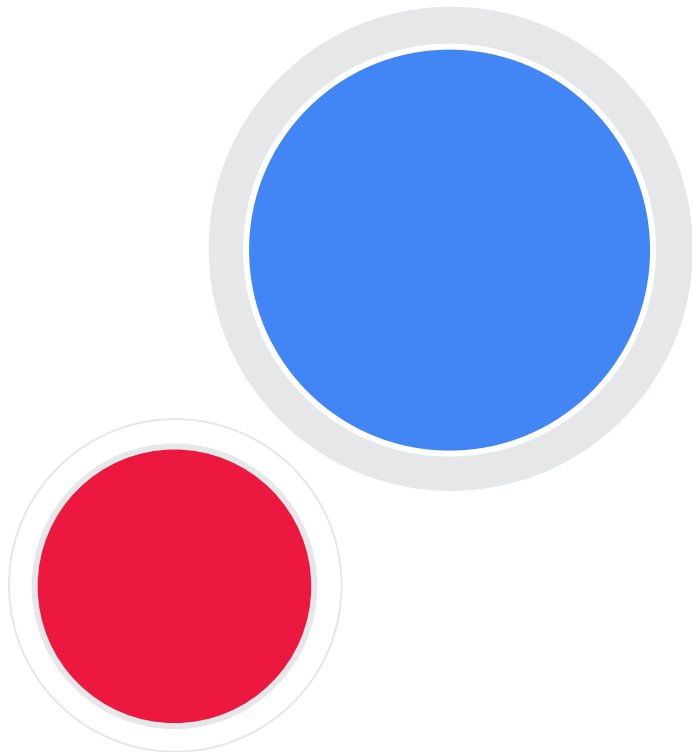


# Objetivo

## Possibilidades

- Colorir textos
- Colorir elementos

Utilizamos cores para dar mais vida, destaque e importância a elementos em nossa página.

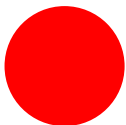




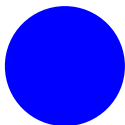
## Cores predefinidas

Por padrão, já temos diversas cores predefinidas que podem ser acessadas diretamente pelo seu nome.

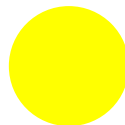
A lista é grande, mas algumas delas são:



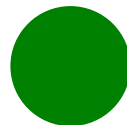
red



blue



yellow



green



## Aplicação das cores

Agora que já vimos alguns nomes, podemos aplicá-las a algum elemento.

Para isso, basta adicionar a propriedade **color** e definir um **valor** para ela.

```
css color: red;
```

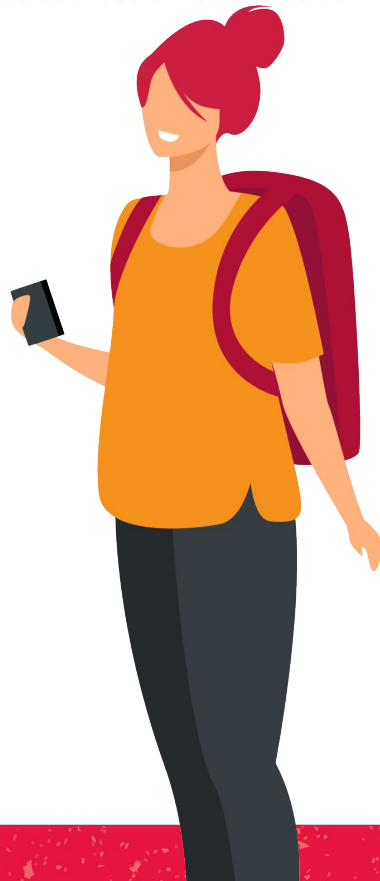


# Mais cores

## Diversas maneiras de representá-las

- Hexadecimais
- RGB

Podemos e devemos usar outras cores além daquelas predefinidas, para isso podemos utilizar algumas maneiras para representá-las...







# Hexadecimais

Uma maneira bem simples de representarmos uma cor é utilizando um código **hexadecimal**.

É um código composto por um **#** seguido por uma **sequência alfanumérica**.

```
css color: #DC143C;
```

*A mesma cor vermelha em formato hexadecimal*



## RGB

Outra maneira de colorirmos os nossos elementos é utilizando o **RGB**.

Para especificarmos uma cor precisamos adicionar o prefixo **rgb** e a cor dentro dos ( )

```
css color: rgb(255, 0, 0);
```

*A mesma cor vermelha em formato RGB*



## Cores predefinidas VS RGB/Hexadecimal

*Ok, mas qual a diferença entre essas duas maneiras?*

E a resposta é simples, **Liberdade**.

Com as cores predefinidas estamos presos em um padrão de cores definidas, mas já com RGB e Hexadecimal, **podemos utilizar qualquer cor existente!**

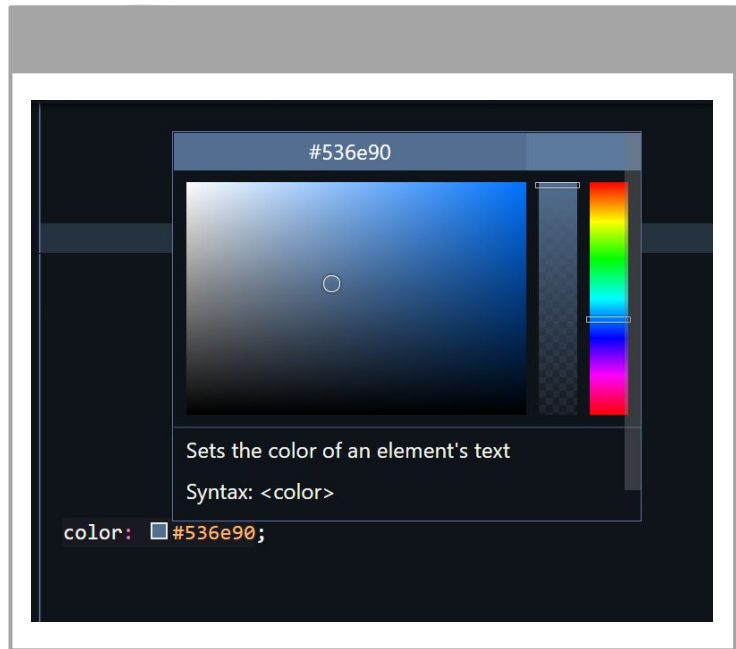




## Preciso decorar todos os códigos? E agora?

Definitivamente não! Existem maneiras bem mais práticas de se obter essas cores nesses formatos.

Uma delas é utilizando o **próprio editor de código para selecionar a cor** que mais te agrada!



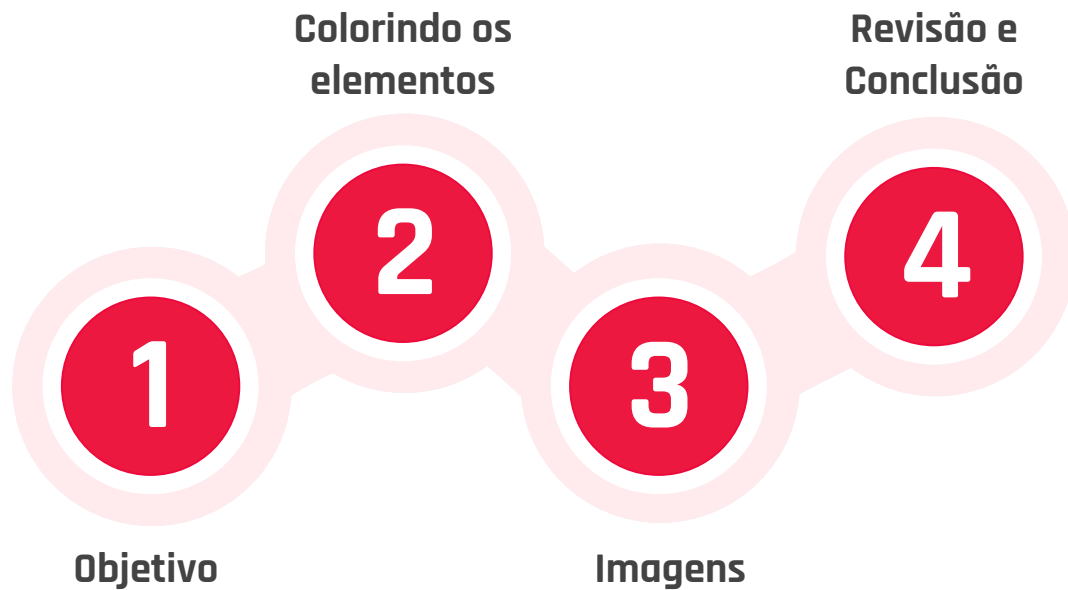
**Hora da revisão**



## Conclusão

- Vimos que é possível **colorir um elemento** via CSS de **diversas maneiras**, utilizando nomes, códigos e funções.
- Através das cores, podemos colorir um texto ou elemento para **atribuir** ou **retirar** sua **importância** dentro da nossa página.
- **Não existe necessidade de decorar** código/função, temos diversas ferramentas para nos ajudarmos nessa tarefa.

## 2 | Background

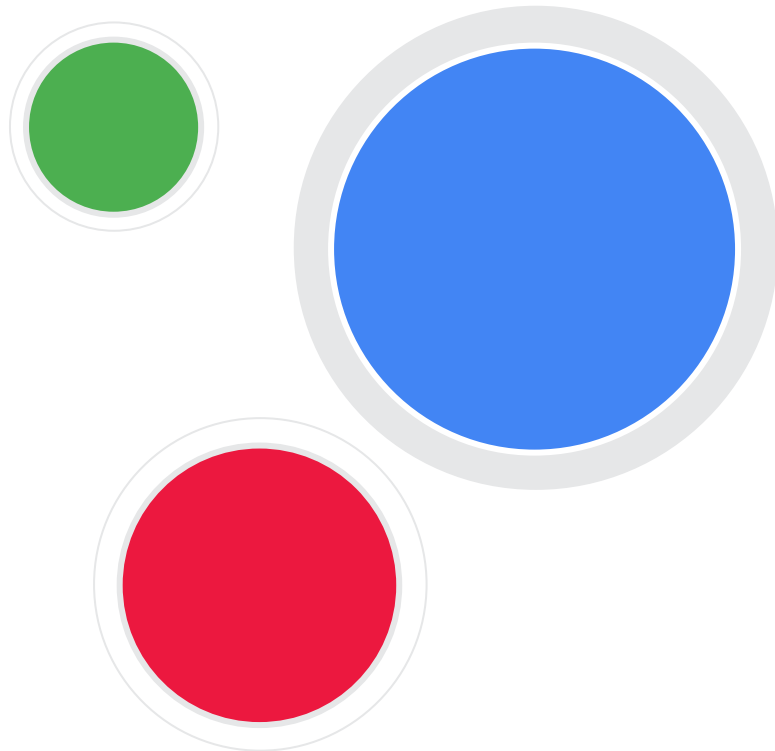






## Objetivo

Agora que descobrimos como as cores funcionam no CSS, vamos aplicá-las em nossos elementos através das propriedades **background**.





## background-color

Ela é responsável por definir a cor de fundo do nosso elemento.

O modo de se utilizar é idêntico à propriedade **color** que vimos anteriormente, onde iremos escrever um nome/código que represente uma cor e o nosso elemento será colorido!

```
css background-color: #DC143C;
```



## Só conseguimos usar cores como background?

Mas é claro que não, nós podemos usar **imagens** também para personalizarmos os nossos elementos.

Para fazer isso, contamos com **diversas propriedades** para nos ajudar com o posicionamento e preenchimento da imagem no nosso elemento.





## background-image

Com essa propriedade, podemos **definir uma imagem como fundo** de nosso elemento. Para isso, precisamos dizer para a propriedade qual imagem queremos usar, e isso é muito fácil.

Temos apenas que informar um link de uma imagem na web ou uma imagem local dentro de: **url(' ')**

```
css background-image: url('imagem.png');
```

*Exemplo com imagem local*



## Posicionando nossa imagem



Existe uma possibilidade da nossa imagem **não ter ficado muito legal** no background do nosso elemento.

Para nos ajudar nisso, temos algumas propriedades de **posicionamento**.



## background-position

Nós podemos definir o posicionamento do nosso background com essa propriedade, precisamos apenas dizer a direção.

Temos a possibilidade de alinhar:

Esquerda (**left**), Topo (**top**), Direita (**right**), Base (**bottom**) e Centro (**center**).

```
css background-position: center;
```

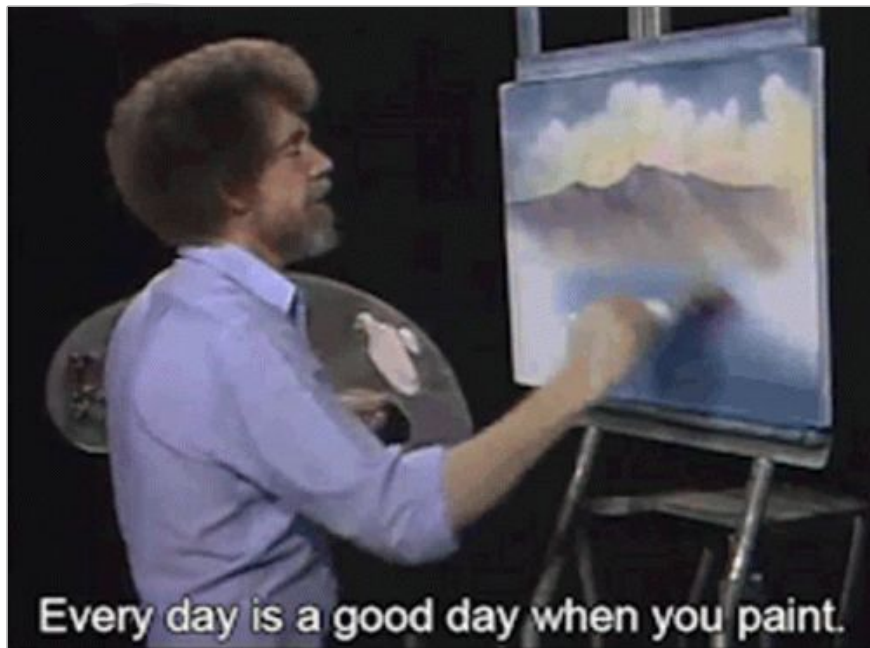
*Exemplo de alinhamento ao centro*



## E muito mais!

Essas são apenas **três** das **oito propriedades** para te ajudar com uso de imagens como background de um elemento.

Explore e divirta-se com as **possibilidades que o CSS te oferece!**



**Hora da revisão**





## Conclusão

- Vimos que é possível **usar uma imagem** como background de um elemento via CSS.
- Podemos **posicionar nossa imagem de fundo** do jeito que acharmos melhor.

# 3 | Fontes

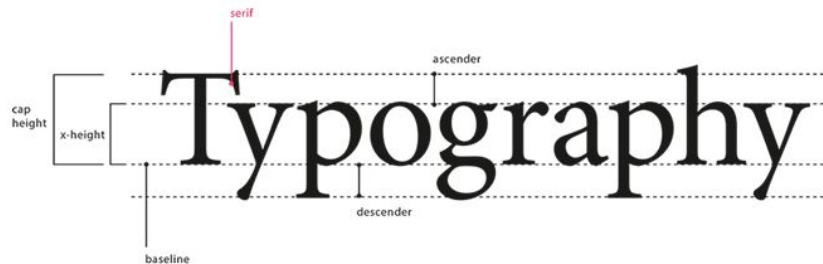




## Objetivo

As fontes são parte **fundamental** de qualquer página web, é essencial ter **uma fonte agradável** para que o público-alvo do seu site **se sinta confortável** lendo alguma informação.

Podemos estilizar o tamanho, tipo, peso e outras diversas propriedades das nossas fontes!





# Fontes Genéricas

**Serif**

# Sans-serif

Geralmente mais utilizada por conta da sua acessibilidade.

**Monospace**



## Definindo uma nova fonte

### font-family

Propriedade que permite adicionar uma nova fonte.

### Fonte genérica

Uma fonte do sistema com esse tipo será utilizada caso a fonte definida não seja encontrada.

css

```
font-family: 'Roboto', sans-serif;
```

### Nome da fonte

Aqui é definida qual fonte você vai utilizar através do próprio nome dela.



## Fontes personalizadas

Podemos incluir fontes personalizadas para deixar a nossa página ainda mais atrativa.

Para isso, vamos utilizar o **Google Fonts**, ele nos fornece diversas fontes prontas para serem utilizadas.

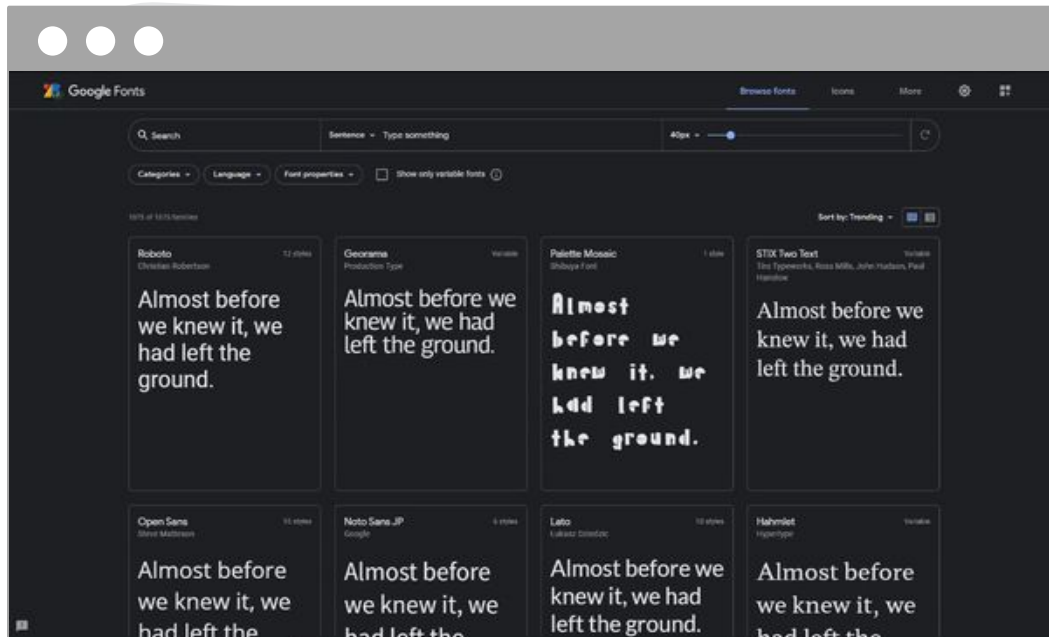




# Google Fonts

O Google Fonts é uma vasta biblioteca de fontes que podemos utilizar ao nosso favor.

Escolha a fonte que mais te agrada para utilizar em sua página.

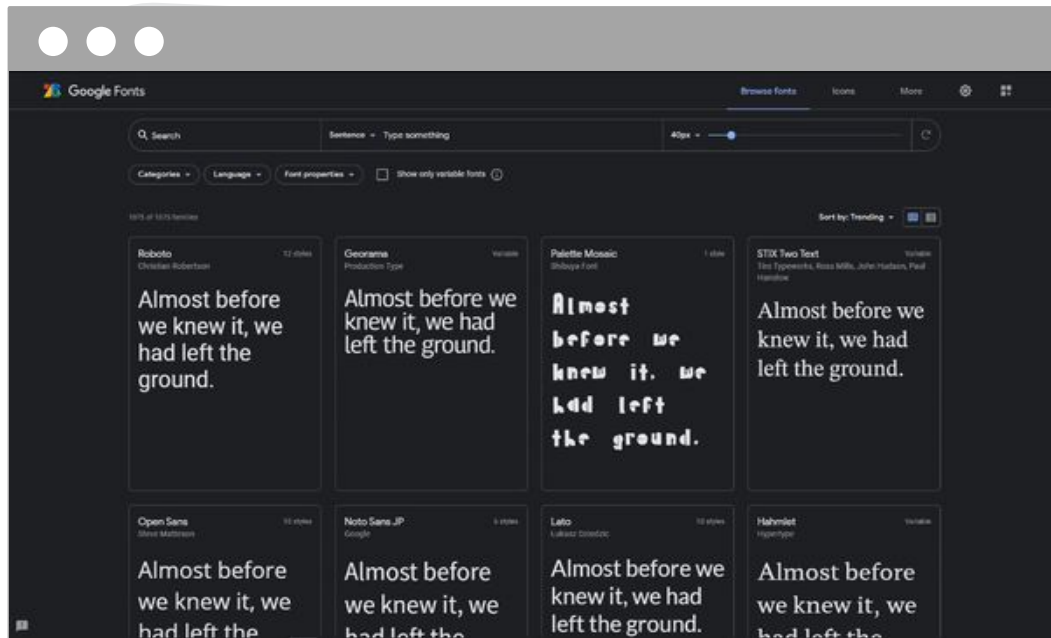






## Importando sua fonte

Entre em [Google Fonts](https://fonts.google.com), escolha uma fonte de seu gosto e clique nela para obtermos mais detalhes sobre a fonte.

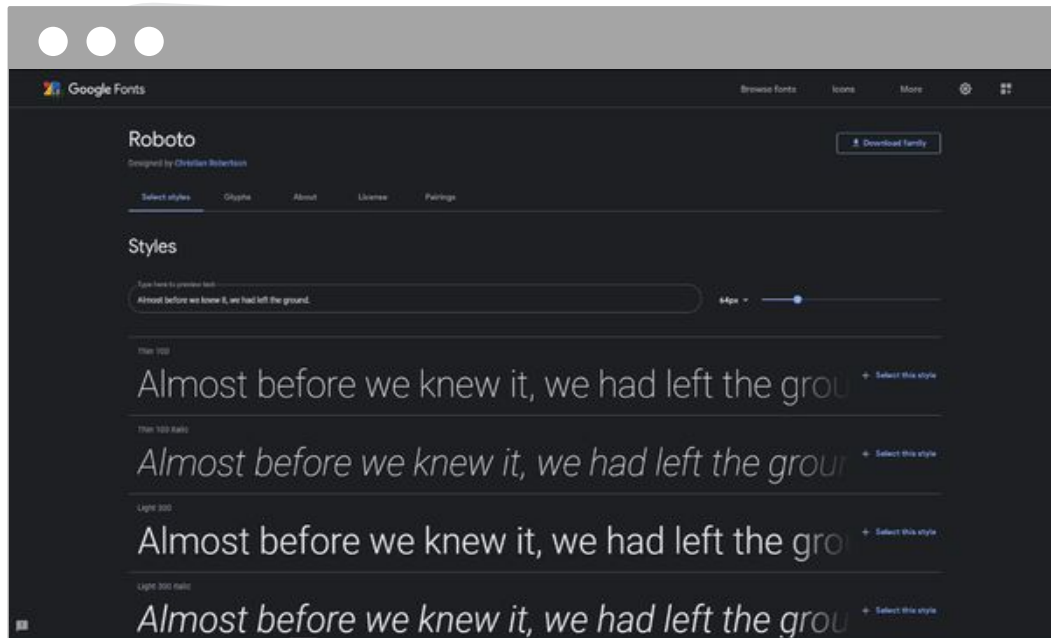




## Importando sua fonte

Entrando na página sobre a fonte, tem opções de fazer download ou utilizarmos direto dos servidores do Google. Vamos utilizar essa segunda opção.

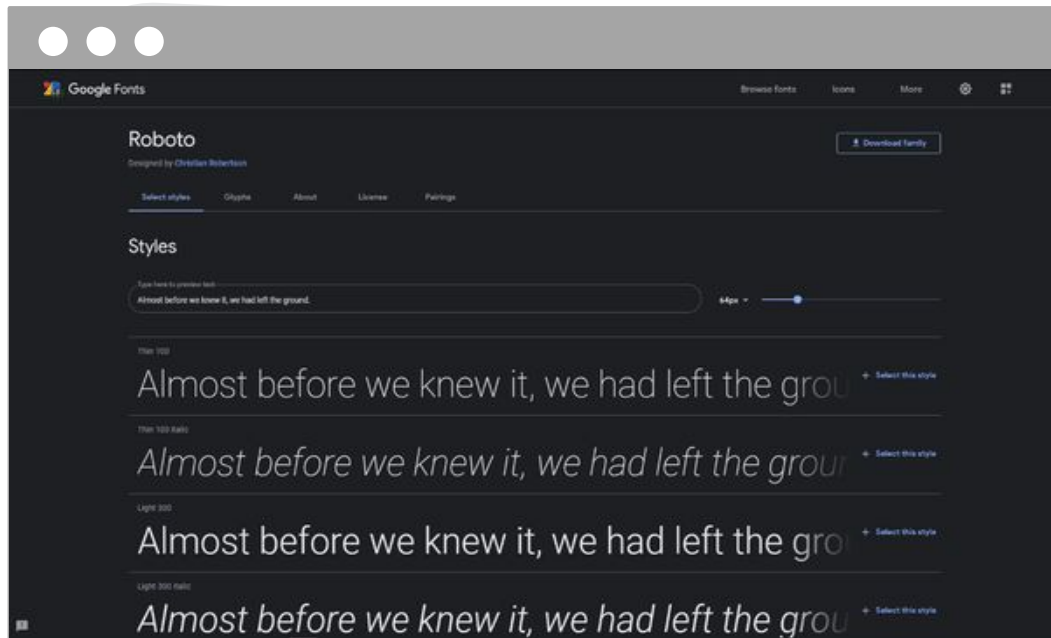
Nessa página, precisamos selecionar quais os pesos nossa fonte irá importar, selecione a 500 que é a padrão.





## Importando sua fonte

Após selecionarmos nossa fonte, um menu na direita será aberto, lá vamos ter duas opções de importações: via HTML e via CSS (representadas por `<link>` e `@import`).

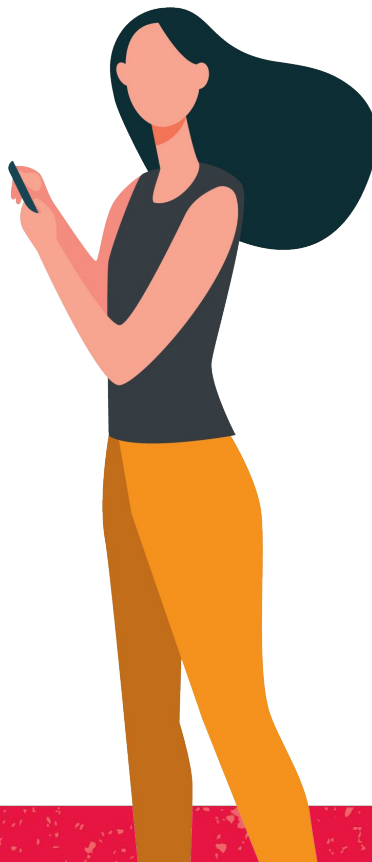




## <link> ou @import ?

Ambas são eficazes naquilo que elas precisam fazer, a principal diferença é que com <link> nós importamos a fonte via HTML e @import via CSS.

Como maneira de organização, podemos fazer as importações de fontes direto do nosso CSS e deixar o HTML para importar coisas mais relevantes como **Folhas de Estilos** e **Scripts**.



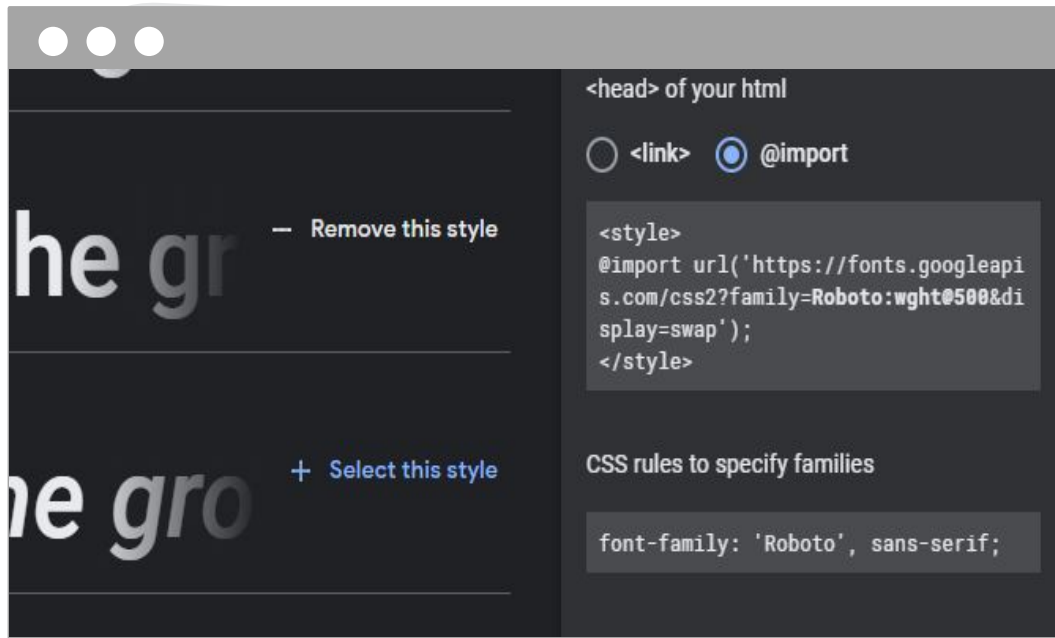




## Importando sua fonte

Abaixo, temos também o **font-family** que ele já nos disponibiliza, então ele pode te ajudar caso fique em dúvida de como declarar no css após importar a fonte.

Lembrando que o **font-family** deve ser utilizado apenas dentro da estilização de algum elemento.





## Utilizando fontes personalizadas

Depois de copiar é só colar em nosso arquivo CSS e nossa fonte já estará funcionando. Você sempre deve colocar o `@import` nas primeiras linhas antes de começar a estilizar os elementos, apenas para garantir que a fonte irá ser importada.

```
css  @import url('www.linkParaFonte.com ou caminho')  
  
    .elemento {  
        font-family: 'Nome da Fonte', sans-serif;  
    }
```

**Hora da revisão**





## Conclusão

- É possível utilizar diferentes fontes.
- Podemos utilizar uma fonte reserva caso a nossa fonte principal não seja encontrada.

DigitalHouse>  
Coding School