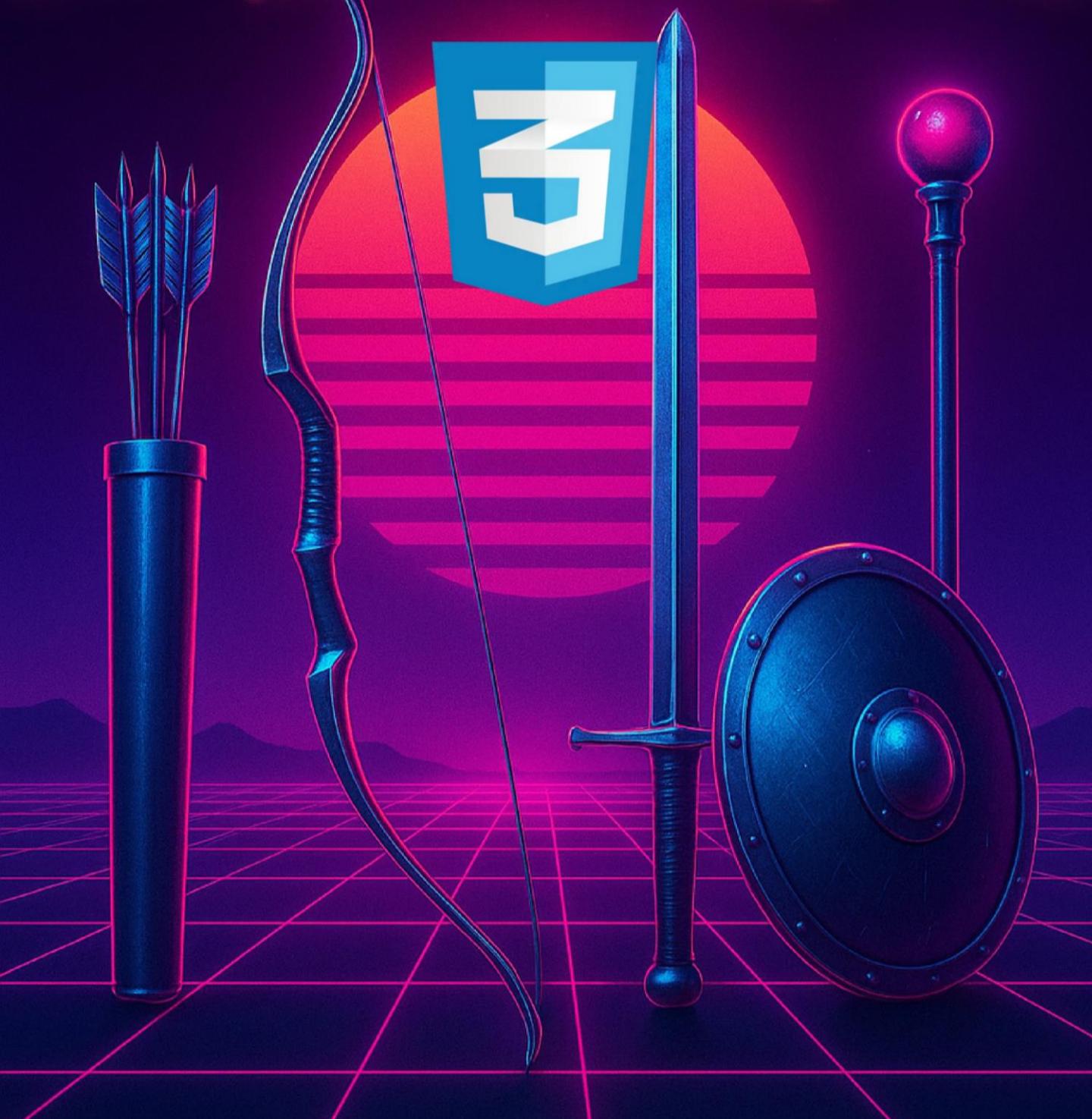


O SENHOR DOS ESTILOS

A SOCIEDADE CSS



IGOR DE SOUZA

01

O Chamado do Front-End:
Entendendo o CSS

O CHAMADO DO FRONT-END: ENTENDENDO O CSS

Todo desenvolvedor inicia sua aventura ao descobrir que a beleza de um site não está apenas no código, mas na forma como ele se apresenta.

E quem controla essa aparência é o CSS – Cascading Style Sheets, ou Folhas de Estilo em Cascata.

- ◆ **O que é CSS?**

O CSS é a linguagem usada para descrever o visual e o estilo dos elementos criados com HTML.

Ele controla cores, tamanhos, fontes, espaçamento, posicionamento, animações e muito mais.

HTML é o “esqueleto” do site.

CSS é o “manto mágico” que dá forma, vida e estética.

Sem CSS, um site é apenas texto bruto e imagens desalinhadas.

Com CSS, surge o design — o equilíbrio entre estrutura e arte.

- ◆ Por que se chama “em cascata”?

O termo “Cascading” vem da forma como o CSS aplica suas regras.

Quando há mais de uma regra afetando o mesmo elemento, o navegador precisa decidir qual delas é mais forte.
Essa hierarquia é chamada de “cascata”.

A ordem de prioridade é:

Estilo Inline (escrito diretamente na tag HTML).

Estilo Interno (escrito dentro da tag <style>).

Estilo Externo (arquivo .css importado).

Estilos padrões do navegador (o que ele aplica por padrão).

Quando dois estilos entram em conflito, vence o que tiver maior prioridade ou for declarado mais recentemente.

- ◆ Três formas de adicionar CSS

1 Inline (na própria tag)

```
<h1 style="color: gold; text-align: center;">Bem-vindo à Sociedade do CSS</h1>
```

Bom para testes rápidos.

Péssimo para projetos grandes (difícil de manter e repetir).

2 Interno (no mesmo arquivo HTML)

```
h1 {  
    color: gold;  
    text-align: center;  
}  
</style>
```

Útil para páginas únicas.

Mistura HTML e CSS, reduz a organização.

3 Externo (arquivo separado)

HTML

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

E no style.css:

```
h1 {  
  color: gold;  
  text-align: center;  
}
```

É o método profissional.

Permite reuso, clareza e manutenção — como guardar cada feitiço em seu próprio grimório.

02

Estrutura da Magia: Como o CSS é escrito

ESTRUTURA DA MAGIA: COMO O CSS É ESCRITO

Cada regra CSS segue um padrão fixo:

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

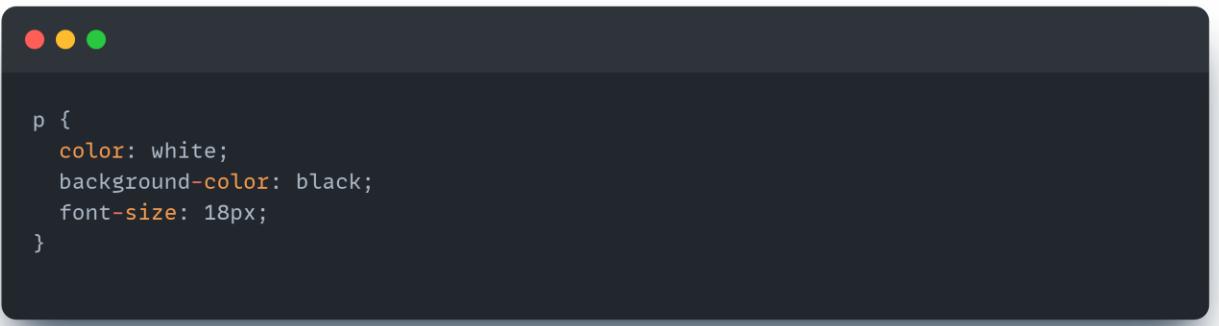
Quebra da estrutura:

Seletor: indica o alvo do feitiço, ou seja, qual elemento será estilizado.

Propriedade: define o que será alterado (cor, fonte, espaçamento, etc).

Valor: diz como será alterado.

Exemplo prático:



```
p {  
  color: white;  
  background-color: black;  
  font-size: 18px;  
}
```

Explicando:

p → todos os parágrafos <p> da página.

color: white; → muda a cor do texto para branco.

background-color: black; → muda o fundo para preto.

font-size: 18px; → define o tamanho da letra.

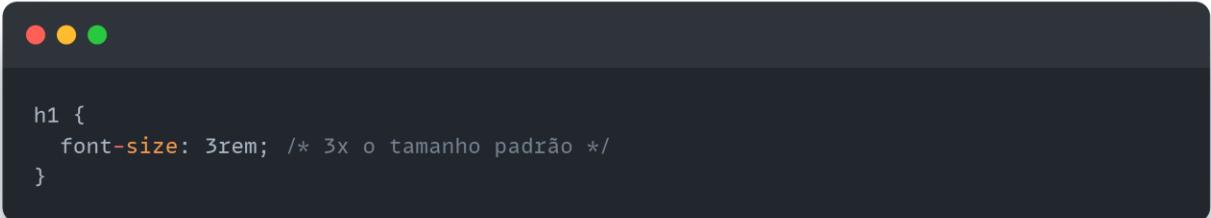
Unidades de medida

CSS usa várias unidades para definir tamanhos:

- **px (pixels): valor fixo, exato.**
- **% (porcentagem): relativo ao tamanho do elemento pai.**

-
- **em**: relativo ao tamanho da fonte do elemento pai.
 - **rem**: relativo à fonte raiz (geralmente o <html>).
 - **vw / vh**: baseadas na largura/altura da janela (viewport).

Exemplo:

A screenshot of a Mac OS X window. The title bar is dark with three colored window control buttons (red, yellow, green) on the left. The main content area contains the following CSS code:

```
h1 {  
    font-size: 3rem; /* 3x o tamanho padrão */  
}
```

03

A Irmandade dos Seletores

A IRMANDADE DOS SELETORES

Os seletores são os guerreiros do CSS, responsáveis por apontar quais elementos receberão os estilos.

- ◆ **Seletor de tipo (ou elemento)**

Aplica o estilo a todas as tags do mesmo tipo:



```
h1 {  
  color: gold;  
}  
p {  
  color: white;  
}
```

Todos os títulos <h1> e parágrafos <p> receberão esses estilos.

- ◆ **Seletor de classe**

As classes permitem criar grupos de elementos com o mesmo estilo.

Sempre começam com um ponto (.).

HTML:

```
<p class="elfo">Legolas é veloz.</p>
<p class="elfo">Tauriel é precisa.</p>
```

CSS:

```
.elfo {
  color: green;
  font-style: italic;
}
```

Todos os elementos com class="elfo" ficam verdes e itálicos.
Você pode aplicar várias classes no mesmo elemento:

```
<p class="elfo arqueiro">Legolas</p>
```

◆ Seletor de ID

IDs são identificadores únicos.

Usam o símbolo # e devem ser únicos por página.

HTML:

```
<p id="mago">Gandalf, o Cinzento.</p>
```

CSS:

```
#mago {  
  color: blue;  
  font-weight: bold;  
}
```

Nunca repita IDs.

Eles são “magias exclusivas”.

◆ Seletor universal

```
* {  
    margin: 0;  
    padding: 0;  
}
```

Afeta todos os elementos.

Usado para “resetar” estilos padrão.

- ◆ Seletores descendentes

Afetam elementos dentro de outros:

```
div p {  
    color: gray;  
}
```

Somente os <p> dentro de <div> serão estilizados.

- ◆ Pseudo-classes e pseudo-elementos

Eles permitem estilizar estados especiais e partes específicas de um elemento.

Exemplos:



```
a:hover {  
    color: gold; /* quando o mouse passa */  
}  
  
p::first-letter {  
    font-size: 200%;  
    color: red;  
}
```

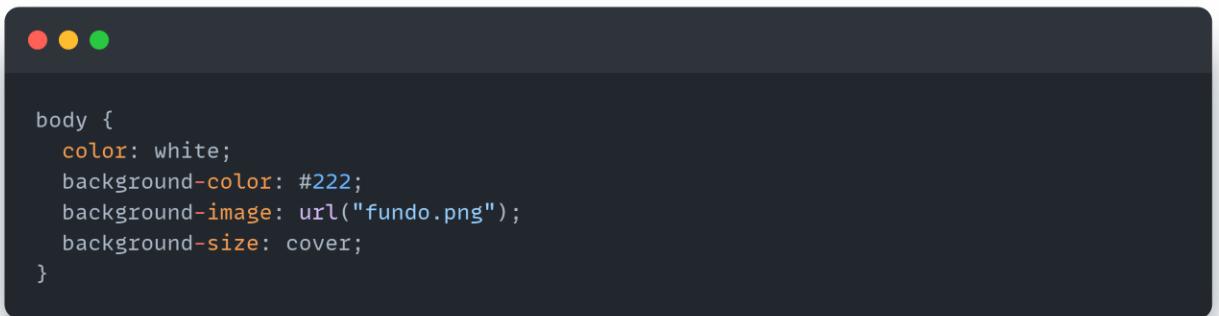
04

As Propriedades Encantadas

AS PROPRIEDADES ENCANTADAS

Agora que sabemos quem será estilizado, vamos aprender o que podemos estilizar.

◆ Cores e fundos



- **color: cor do texto.**
- **background-color: cor do fundo.**
- **background-image: define uma imagem de fundo.**
- **background-size: ajusta a imagem (cover, contain, auto).**

◆ Fontes e textos

```
p {  
    font-family: "Poppins", sans-serif;  
    font-size: 16px;  
    font-weight: 400;  
    text-align: justify;  
    line-height: 1.5;  
}
```

- **font-family:** tipo de fonte.
- **font-size:** tamanho da letra.
- **font-weight:** espessura (100 a 900, ou normal/bold).
- **text-align:** alinhamento.
- **line-height:** espaçamento entre linhas.

◆ Margem e padding

Essenciais para espaçamento e organização.

```
div {  
    margin: 20px;  
    padding: 10px;  
}
```

margin: espaço fora da borda (distância entre elementos).

padding: espaço dentro da borda (entre o conteúdo e a borda).

Pense no margin como o espaço entre casas, e o padding como o espaço dentro da casa.



```
.card {  
  border: 2px solid gold;  
  border-radius: 10px;  
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);  
}
```

border: cria borda (largura, estilo, cor).

border-radius: arredonda cantos.

box-shadow: cria sombra.

05

Layouts: Estruturando o Reino

LAYOUTS: ESTRUTURANDO O REINO

Organizar elementos é essencial.

Aqui entram os sistemas **display**, **flexbox**, **grid** e **position**.

◆ **Display**

```
div {  
  display: block;  
}  
span {  
  display: inline;  
}
```

- **block**: ocupa toda a largura disponível.
- **inline**: ocupa apenas o espaço necessário.
- **flex e grid**: modos modernos de layout.

◆ **Flexbox – O guerreiro disciplinado**

```
.container {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}
```

-
- **justify-content: alinha horizontalmente.**
 - **align-items: alinha verticalmente.**
 - **flex-direction: muda o eixo (linha ou coluna).**
 - **gap: define o espaçamento entre os itens.**
- ◆ Grid – O arquiteto da Terra Média

```
.grid {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
  gap: 20px;  
}
```

Cria uma grade de colunas.

Cada 1fr representa uma fração do espaço disponível.

◆ Position – Controle absoluto

```
.elemento {  
  position: absolute;  
  top: 50px;  
  left: 20px;  
}
```

-
- **static:** padrão (sem posicionamento).
 - **relative:** desloca em relação à posição normal.
 - **absolute:** fixa em relação ao elemento pai.
 - **fixed:** fixa na tela (mesmo rolando).
 - **sticky:** combina comportamento fixo e relativo.

06

Responsividade, Transições e Boas Práticas

RESPONSIVIDADE, TRANSIÇÕES E BOAS PRÁTICAS

- ◆ Responsividade com media queries



```
@media (max-width: 768px) {  
  body {  
    background-color: black;  
  }  
}
```

Adapta o design para telas menores (celulares, tablets).

- ◆ Transições suaves



```
.botao {  
  background-color: gold;  
  transition: background-color 0.3s ease;  
}  
  
.botao:hover {  
  background-color: orange;  
}
```

-
- **transition**: define o tempo e tipo da animação.
 - **:hover**: muda o estilo quando o mouse passa.
 - ◆ Organização e boas práticas
 - **Comente o código**



- **Use nomes descritivos.**
.menu-principal, .botao-primario
- **Evite repetir regras — use classes reutilizáveis.**
- **Organize o código por seções (cores, textos, layout).**
- **Sempre valide seu CSS antes de publicar.**

Epílogo

O Retorno do Dev

Você agora domina os fundamentos da magia dos estilos.

Com CSS, você é capaz de controlar a forma, a harmonia e o equilíbrio da web.

Mas lembre-se:

“Nem todo código que brilha é dourado.” — um bom design vem da prática e do entendimento do usuário.

Seu próximo passo:

Aprender animações complexas (keyframes).

Dominar variáveis CSS, pseudo-classes avançadas, transforms, pré-processadores (SASS, LESS).

E, acima de tudo: criar, testar e experimentar.

AGRADECIMENTOS

Este ebook foi criado utilizando IA como base, para o desafio da plataforma DIO, relacionado ao curso Universia - Fundamentos de IA Generativa.

Obrigado por ler até o momento!

Segue abaixo os meus contatos.

<https://github.com/IgorSouzDEV>

linkedin.com/in/igor-de-souza-aguiar-1259a9168/