

# Glossário Completo de CSS para Iniciantes

Este glossário foi elaborado para iniciantes que desejam aprender **CSS (Cascading Style Sheets)** e entender como estilizar sites e aplicações web. Cada conceito vem acompanhado de exemplos práticos para facilitar o aprendizado.

---

## ✦ O que é CSS?

CSS é uma linguagem usada para definir o **estilo** de páginas HTML: cores, fontes, espaçamentos, layouts e muito mais.

---

## ✦ Seletores

São utilizados para selecionar elementos HTML que receberão estilos.

```
/* Seleciona todos os <p> */
p {
  color: red;
}

/* Seleciona um elemento pelo id */
#titulo {
  font-size: 24px;
}

/* Seleciona elementos pela classe */
.card {
  background-color: #f0f0f0;
}
```

---

## ✦ Propriedades Básicas

- `color`: cor do texto
- `background-color`: cor de fundo
- `font-size`: tamanho da fonte
- `font-family`: tipo da fonte
- `text-align`: alinhamento do texto
- `padding`: espaçamento interno
- `margin`: espaçamento externo
- `border`: borda do elemento

```
h1 {
```

```
color: blue;
font-size: 32px;
margin-bottom: 20px;
}
```

---

## ✦ Modelo de Caixa (Box Model)

Todo elemento HTML é uma **caixa** composta por:

```
[ margin ]
[ border ]
[ padding ]
[ conteúdo ]
```

Exemplo:

```
div {
  width: 300px;
  padding: 20px;
  border: 2px solid black;
  margin: 10px;
}
```

---

## ✦ Classes e IDs

- `.classe` → para múltiplos elementos
- `#id` → para um único elemento

```
.container {
  max-width: 1200px;
}
```

---

## ✦ Unidades de Medida

- `px`: pixels (fixo)
- `%`: porcentagem (relativo)
- `em` / `rem`: relativo ao tamanho da fonte

```
p {
  font-size: 16px;
  padding: 2em;
}
```

---

## ✦ Cores

- Nome: `red`
- Hexadecimal: `#ff0000`

- RGB: `rgb(255, 0, 0)`
  - RGBA (com opacidade): `rgba(255, 0, 0, 0.5)`
- 

## ✦ Pseudo-classes

Estilizam elementos em determinados estados.

```
a:hover {  
  color: green;  
}
```

---

## ✦ Display

Controla o comportamento do elemento na página.

- `block` → ocupa toda a linha
- `inline` → na mesma linha de outros elementos
- `flex` → layout flexível

```
.container {  
  display: flex;  
}
```

---

## ✦ Flexbox

Sistema moderno de layout.

```
.container {  
  display: flex;  
  justify-content: center; /* Alinhamento horizontal */  
  align-items: center;     /* Alinhamento vertical */  
}
```

---

## ✦ Grid

Layout em grade.

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

---

## ✦ Posição (Position)

Define como o elemento será posicionado.

- static (padrão)
- relative
- absolute
- fixed
- sticky

```
.box {  
  position: absolute;  
  top: 50px;  
  left: 100px;  
}
```

---

## ✦ Responsividade (Media Queries)

Permite criar layouts adaptáveis para diferentes tamanhos de tela.

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

---

## ✦ Variáveis CSS

Permitem armazenar valores reutilizáveis.

```
:root {  
  --cor-principal: #3498db;  
}  
  
button {  
  background-color: var(--cor-principal);  
}
```

---

## ✦ Animações e Transições

**Transições:**

```
button {  
  background-color: blue;  
  transition: background-color 0.3s;  
}  
  
button:hover {  
  background-color: red;  
}
```

```
}
```

### Animações:

```
@keyframes girar {  
  from {  
    transform: rotate(0deg);  
  }  
  to {  
    transform: rotate(360deg);  
  }  
}  
  
img {  
  animation: girar 2s infinite;  
}
```

---

## ✦ Z-index

Controla a sobreposição de elementos.

```
.elemento {  
  position: absolute;  
  z-index: 10;  
}
```

---

## ✦ Importação de Fontes (Google Fonts)

```
@import  
url('https://fonts.googleapis.com/css2?family=Roboto&display=swap');  
  
body {  
  font-family: 'Roboto', sans-serif;  
}
```

---

## ✦ Bibliotecas Populares

- **Bootstrap** — Framework CSS para sites responsivos
- **Tailwind CSS** — Estilização utilitária
- **Animate.css** — Animações prontas