

HTML 5 E CSS 3



Introdução ao CSS3

O que é CSS e como aplicá-lo

1. Introdução

O CSS (Cascading Style Sheets), ou Folhas de Estilo em Cascata, é uma linguagem usada para descrever a apresentação visual de documentos HTML. Enquanto o HTML define a estrutura e o conteúdo da página, o CSS é responsável pela aparência: cores, fontes, espaçamentos, posicionamentos e estilos visuais em geral. Sua separação do HTML permite maior controle, reaproveitamento de estilos e organização do código, além de facilitar a manutenção e a escalabilidade de projetos web.

2. Diferença entre HTML e CSS

O HTML e o CSS trabalham em conjunto na construção de uma página web, mas possuem funções distintas e complementares:

- **HTML (HyperText Markup Language):** estrutura o conteúdo da página. É usado para definir textos, imagens, listas, links, formulários, títulos e outras partes do conteúdo.
- **CSS (Cascading Style Sheets):** estiliza esse conteúdo. Controla aspectos visuais como layout, cores, tamanhos de fonte, margens, espaçamentos e posicionamentos.

Exemplo de diferença:

<!-- HTML define um parágrafo -->

<p>Bem-vindo ao meu site!</p>

<!-- CSS define como esse parágrafo será exibido -->

<style>

**p {
color: blue;
font-size: 16px;
}**

</style>

Enquanto o HTML apenas apresenta o texto, o CSS define que esse texto será azul e com um tamanho de 16 pixels. Essa separação promove um design mais limpo e adaptável, especialmente em projetos maiores.

3. Formas de usar CSS

Há três principais formas de aplicar CSS em um documento HTML: inline, interna e externa. Cada uma possui finalidades e contextos de uso específicos.

3.1 CSS Inline

O CSS inline é aplicado diretamente em um elemento HTML, usando o atributo style. É útil para estilizações pontuais e rápidas, mas não é recomendado para projetos grandes, pois dificulta a manutenção.

Exemplo:

```
<p style="color: red; font-weight: bold;">Texto em vermelho e  
negrito</p>
```

Neste exemplo, o CSS está embutido no próprio elemento `<p>`. A desvantagem é a repetição de estilos quando há muitos elementos com aparência semelhante.

3.2 CSS Interno

O CSS interno é colocado dentro da própria página HTML, dentro da tag `<style>`, geralmente dentro do `<head>`. É mais organizado que o inline e permite aplicar estilos a múltiplos elementos.

Exemplo:

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

Portal
IDEA
.com.br

Esse método é indicado para páginas simples, mas ainda mantém o estilo acoplado ao conteúdo, o que dificulta a reutilização em vários arquivos.

3.3 CSS Externo

O CSS externo é definido em um arquivo .css separado, que é vinculado ao HTML por meio da tag <link>. Esse método é o mais recomendado para projetos profissionais e escaláveis.

Exemplo do HTML:

```
<head>
```

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

```
</head>
```

Arquivo estilos.css:

```
css
```

```
CopiarEditar
```

```
body {
```

```
background-color: #f5f5f5;
```

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

```
}
```

A principal vantagem do CSS externo é a reutilização: um único arquivo de estilos pode ser usado em diversas páginas, promovendo consistência visual e facilidade de manutenção.

4. Sintaxe básica de CSS

O CSS é baseado em uma sintaxe simples e legível, composta por **seletores**, **propriedades** e **valores**. A estrutura geral é:

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

4.1 Seletores

Os **seletores** definem a quais elementos HTML o estilo será aplicado. Podem ser simples (por nome da tag) ou mais complexos (por classe, id, ou combinadores).

- **Por tag:** aplica a todos os elementos de um tipo.

```
p {  
  
    color: blue;  
  
}
```

- **Por classe:** usa . e permite aplicar o mesmo estilo a múltiplos elementos.

```
.destaque {  
  
    background-color: yellow;  
  
}
```

- **Por id:** usa # e aplica a um único elemento.

css

CopiarEditar

```
#titulo {  
  
    font-size: 24px;  
  
}
```

4.2 Propriedades

As **propriedades** definem quais aspectos visuais serão alterados, como cor, espaçamento, fonte, bordas, entre outros.

Exemplos comuns:

- `color`: define a cor do texto.
- `background-color`: define a cor de fundo.
- `font-size`: controla o tamanho da fonte.
- `margin`: define o espaço externo do elemento.
- `padding`: define o espaço interno entre o conteúdo e a borda.

4.3 Valores

Cada propriedade recebe um **valor** correspondente. Os valores podem ser números com unidades (px, %, em), palavras-chave (bold, center), cores (red, #ff0000), entre outros.

Exemplo completo:

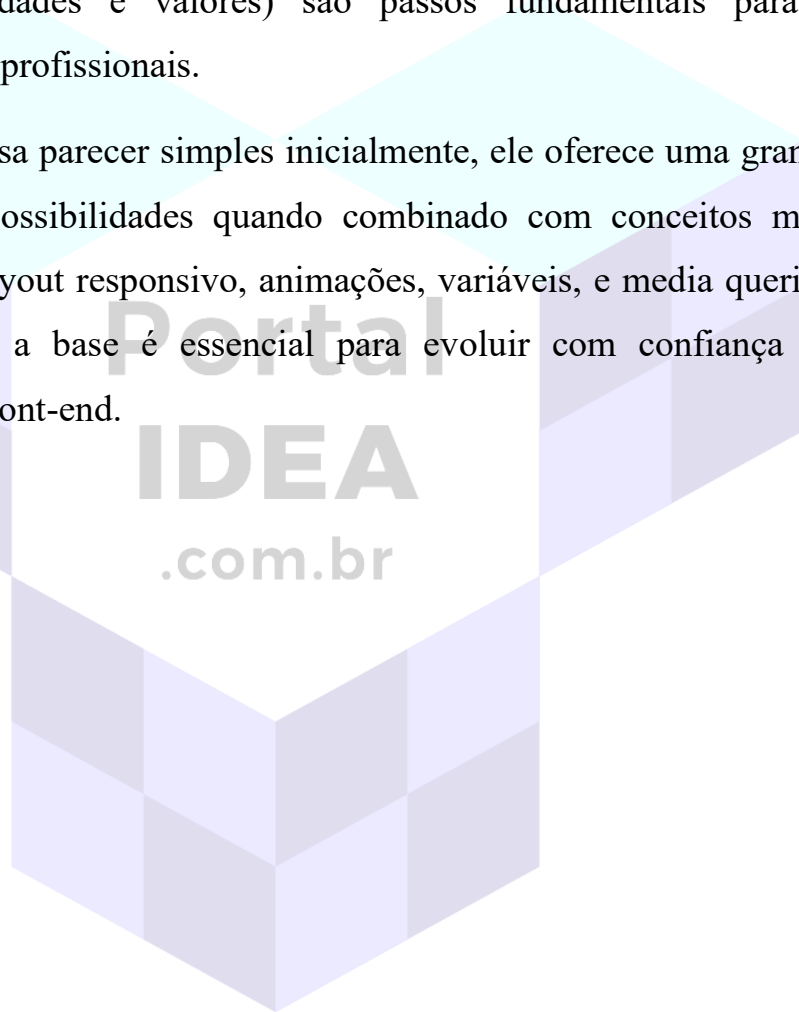
```
h1 {  
  
  color: navy;  
  
  font-size: 32px;  
  
  text-align: center;  
  
}
```

Esse código define que todos os elementos `<h1>` terão cor azul escura, fonte de 32 pixels e alinhamento centralizado.

5. Considerações finais

O CSS é uma ferramenta indispensável no desenvolvimento web moderno. Ele permite que desenvolvedores criem interfaces visuais atrativas, organizadas e acessíveis, separando de forma clara o conteúdo da sua apresentação. Entender as diferenças entre HTML e CSS, conhecer as três formas principais de aplicação de estilos e dominar a sintaxe básica (seletores, propriedades e valores) são passos fundamentais para a construção de sites profissionais.

Embora o CSS possa parecer simples inicialmente, ele oferece uma grande profundidade de possibilidades quando combinado com conceitos mais avançados como layout responsivo, animações, variáveis, e media queries. Portanto, dominar a base é essencial para evoluir com confiança no desenvolvimento front-end.



Referências Bibliográficas

- DUCKETT, Jon. *HTML and CSS: Design and Build Websites*. Wiley, 2011.
- FREEMAN, Elisabeth; ROBSON, Eric. *Use a Cabeça! HTML e CSS*. Alta Books, 2015.
- MDN Web Docs – Mozilla Developer Network. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>
- W3Schools. *CSS Tutorial*. Disponível em: <https://www.w3schools.com/css>
- CASTRO, Elizabeth; HYSLOP, Bruce. *HTML5 e CSS3: Guia Prático e Visual*. Alta Books, 2013.

Portal
IDEA
.com.br

Seletores e Cores em CSS

1. Introdução

O CSS (Cascading Style Sheets) permite a estilização visual de páginas HTML com precisão e flexibilidade. Dois dos fundamentos mais importantes dessa linguagem são os **seletores**, que determinam *quem será estilizado*, e os **valores de cor**, que definem *como esse elemento será visualmente apresentado*. Compreender a lógica dos seletores e os diferentes modos de aplicar cores é essencial para qualquer desenvolvedor web que deseje controlar a aparência e a hierarquia dos estilos de forma eficaz.

2. Seletores simples: por tag, classe e ID

Os seletores são instruções que informam ao navegador quais elementos HTML devem ser estilizados. Os **seletores simples** são os mais utilizados e acessíveis, permitindo o controle direto de elementos com base em sua marcação no HTML.

2.1 Seletor por tag (elemento)

Este seletor aplica o estilo a todos os elementos de um tipo específico.

Exemplo:

```
p {  
  
    font-size: 16px;  
  
    line-height: 1.5;  
  
}
```

Neste exemplo, todos os parágrafos <p> da página receberão as propriedades definidas.

2.2 Seletor por classe

A classe é definida no HTML com o atributo class e é precedida por um ponto (.) no CSS. Pode ser aplicada a múltiplos elementos, permitindo a reutilização de estilos.

HTML:

```
<p class="destaque">Texto em destaque.</p>
```

CSS:

```
.destaque {  
    background-color: yellow;  
}
```

As classes são ideais para aplicar estilos a grupos de elementos que compartilham características visuais.

2.3 Seletor por ID

O ID é único dentro do documento HTML. Ele é definido com o atributo id no HTML e precedido por uma cerquilha (#) no CSS.

HTML:

```
<h1 id="titulo-principal">Bem-vindo</h1>
```

CSS:

```
#titulo-principal {  
    color: navy;  
}
```

IDs são utilizados principalmente para aplicar estilos únicos e para manipulação via JavaScript. Entretanto, seu uso excessivo em CSS deve ser evitado, pois dificulta a reutilização e a manutenção do código.

3. Cores no CSS: nomes, valores hexadecimais, rgb() e rgba()

A definição de cor é uma das propriedades mais utilizadas no CSS, seja para textos, fundos ou bordas. Existem diferentes formas de representar cores, e a escolha entre elas depende de fatores como controle de opacidade, padronização visual e legibilidade do código.

3.1 Cores por nome

O CSS reconhece cerca de 140 nomes de cores, como red, blue, green, black, white, entre outros.

Exemplo:

```
h1 {  
  
    color: green;  
  
}
```

Essa abordagem é simples, mas limitada, pois nem todos os tons podem ser representados por nomes.

3.2 Valores hexadecimais

O formato hexadecimal usa uma combinação de seis dígitos precedidos por #, representando as cores em componentes RGB (vermelho, verde, azul).

Exemplo:

```
body {  
  
    background-color: #f0f0f0;
```

```
}
```

Nesse caso, #f0f0f0 é um tom de cinza claro. Cada par de dígitos varia de 00 a FF, representando de 0 a 255 em base decimal.

3.3 rgb() – Red, Green, Blue

Essa função permite definir a cor diretamente em valores decimais de 0 a 255.

Exemplo:

```
p {  
  color: rgb(0, 128, 255);  
}
```

Aqui, o texto será exibido em azul claro. A vantagem do rgb() é a clareza na definição dos componentes da cor.

3.4 rgba() – RGB com opacidade

O rgba() funciona como o rgb(), mas inclui um quarto parâmetro: a opacidade (ou canal alfa), que varia de 0 (completamente transparente) a 1 (totalmente opaco).

Exemplo:

```
div {  
  background-color: rgba(255, 0, 0, 0.5);  
}
```

Isso define um fundo vermelho com 50% de transparência. Esse formato é ideal para criar efeitos de sobreposição ou transições suaves.

4. Agrupamento e hierarquia de seletores

A organização e hierarquia dos seletores no CSS são fundamentais para evitar conflitos, tornar o código mais limpo e garantir o funcionamento correto das regras.

4.1 Agrupamento de seletores

É possível aplicar o mesmo estilo a diferentes seletores de uma só vez, separando-os por vírgulas.

Exemplo:

```
h1, h2, h3 {  
    font-family: 'Arial', sans-serif;  
}
```

Com isso, todos os títulos principais terão a mesma fonte.

4.2 Hierarquia (especificidade)

A hierarquia de seletores determina **qual regra será aplicada** quando múltiplas regras entrarem em conflito. A ordem de especificidade é:

1. **Estilo inline** (mais específico)
2. **ID**
3. **Classe, atributo e pseudo-classes**
4. **Elemento (tag)**

Exemplo de conflito:

```
<p id="mensagem" class="alerta">Texto de exemplo</p>
```

```
p {  
    color: black;
```

```
}
```

```
.alerta {  
  
  color: orange;  
  
}
```

```
#mensagem {  
  
  color: red;  
  
}
```

Neste caso, o texto será exibido em **vermelho**, pois o seletor por ID tem maior peso que o seletor por classe ou por tag.

4.3 Hierarquia de aninhamento

Também é possível usar seletores combinados para aplicar estilos apenas a elementos que estejam dentro de outros.

Exemplo:

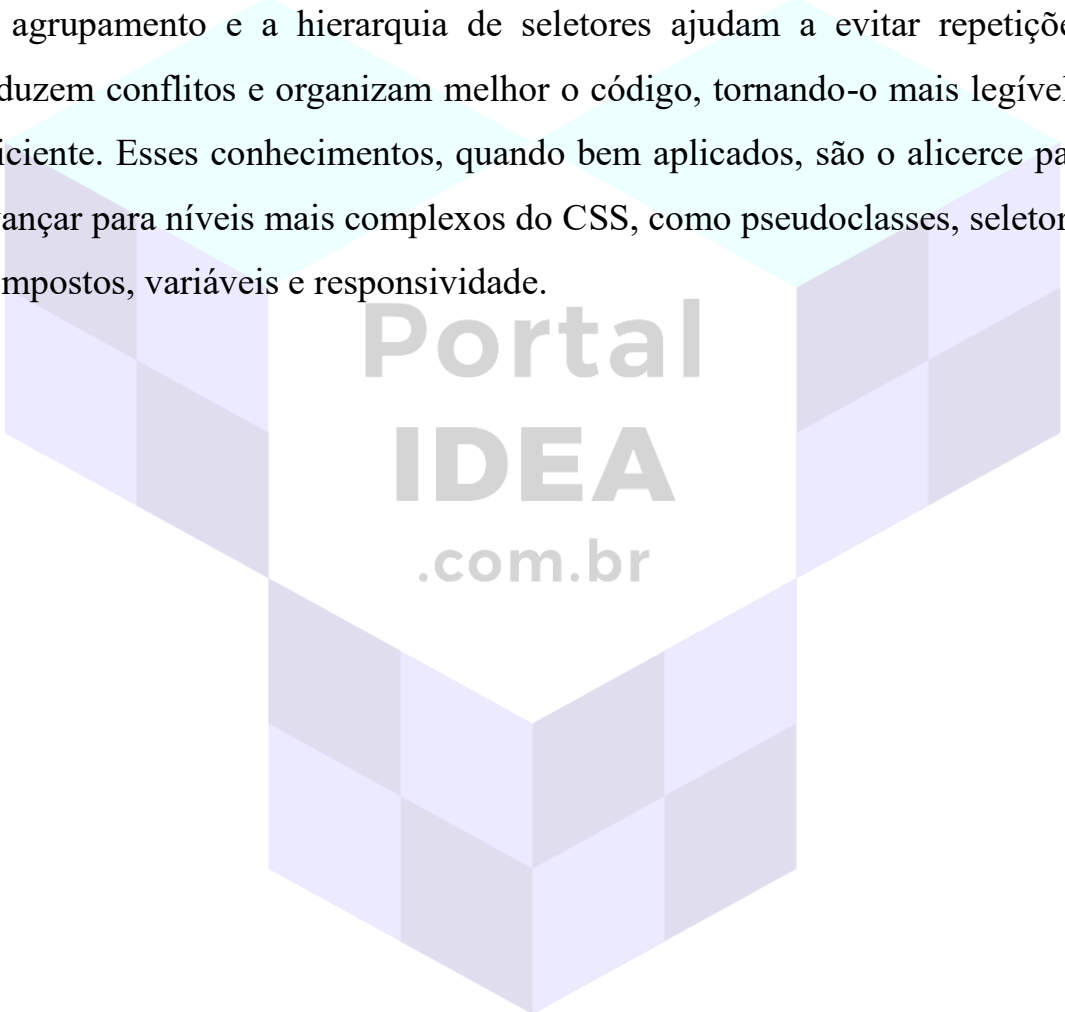
```
article p {  
  
  color: gray;  
  
}
```

Neste caso, apenas os parágrafos que estiverem dentro de um <article> serão afetados.

5. Considerações finais

Dominar os **seletores** e as **formas de definição de cores** é essencial para o trabalho com CSS. Os seletores simples — por tag, classe e ID — são as bases sobre as quais se constrói a maioria dos estilos. Já as cores, fundamentais para a estética e a usabilidade, podem ser aplicadas com diferentes formatos, que variam conforme a necessidade do projeto.

O agrupamento e a hierarquia de seletores ajudam a evitar repetições, reduzem conflitos e organizam melhor o código, tornando-o mais legível e eficiente. Esses conhecimentos, quando bem aplicados, são o alicerce para avançar para níveis mais complexos do CSS, como pseudoclasses, seletores compostos, variáveis e responsividade.

A logo for Portal IDEA .com.br is centered on the page. It features the text 'Portal' in a large, light gray sans-serif font, with 'IDEA' in a bold, light gray sans-serif font directly below it, and '.com.br' in a smaller, light gray sans-serif font at the bottom. The text is overlaid on a background of a large, light blue hexagon that is composed of several smaller, darker blue hexagons arranged in a honeycomb pattern.

Portal
IDEA
.com.br

Referências Bibliográficas

- DUCKETT, Jon. *HTML and CSS: Design and Build Websites*. Wiley, 2011.
- FREEMAN, Elisabeth; ROBSON, Eric. *Use a Cabeça! HTML e CSS*. Alta Books, 2015.
- MDN Web Docs – Mozilla Developer Network. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>
- W3Schools. *CSS Tutorial*. Disponível em: <https://www.w3schools.com/css>
- CASTRO, Elizabeth; HYSLOP, Bruce. *HTML5 e CSS3: Guia Prático e Visual*. Alta Books, 2013.

Portal
IDEA
.com.br

Estilizando Texto e Plano de Fundo com CSS

1. Introdução

O CSS (Cascading Style Sheets) oferece um amplo conjunto de propriedades que permitem controlar a aparência visual de elementos em uma página web. Entre as mais utilizadas estão aquelas que estilizam **textos** e **planos de fundo**, além das propriedades que definem **espaçamentos internos (padding)**, **espaçamentos externos (margin)** e **bordas**. Essas propriedades são fundamentais para a criação de layouts organizados, legíveis e visualmente agradáveis.

2. Estilizando textos com CSS

O texto é um dos principais componentes de qualquer página web. Torná-lo legível, atrativo e bem posicionado é essencial para uma boa experiência do usuário. O CSS permite personalizar diversos aspectos do texto: fonte, tamanho, estilo, peso, espaçamento e alinhamento.

2.1 Fontes

A propriedade font-family define a família de fontes que será aplicada ao texto.

Exemplo:

```
p {  
  
    font-family: 'Arial', sans-serif;  
  
}
```

O uso de múltiplas fontes como fallback é comum, garantindo que, caso a primeira fonte não esteja disponível, uma alternativa seja usada.

2.2 Tamanhos

A propriedade font-size define o tamanho da fonte. Os valores podem ser em px, %, em, rem, entre outras unidades.

Exemplo:

```
h1 {  
  font-size: 32px;  
}
```

O uso de unidades relativas como em e rem é recomendado em projetos responsivos, pois se adaptam melhor a diferentes dispositivos.

2.3 Estilos e peso

- font-style: define se a fonte será normal ou itálica.
- font-weight: determina o peso (espessura), variando de normal, bold ou valores numéricos como 400, 700.

Exemplo:

```
strong {  
  font-weight: bold;  
}  
  
em {  
  font-style: italic;  
}
```

2.4 Alinhamentos

O alinhamento do texto é controlado com a propriedade `text-align`, que pode assumir valores como `left`, `right`, `center` e `justify`.

Exemplo:

```
p {  
    text-align: justify;  
}
```

Outras propriedades úteis para o controle visual do texto incluem `line-height` (altura da linha), `letter-spacing` (espaçamento entre letras) e `text-transform` (transformações como maiúsculas).

3. Alterando plano de fundo

O plano de fundo de um elemento pode ser estilizado com cores sólidas, imagens ou até gradientes. O CSS fornece propriedades específicas para esse fim, como `background-color` e `background-image`.

3.1 Cor de fundo

A propriedade `background-color` define a cor de fundo de um elemento.

Exemplo:

```
body {  
    background-color: #f9f9f9;  
}
```

Pode-se usar nomes de cores, valores hexadecimais, RGB ou HSL.

3.2 Imagem de fundo

A propriedade background-image permite inserir uma imagem como fundo.

Exemplo:

```
div.banner {  
  
  background-image: url('banner.jpg');  
  
}
```

É comum utilizar outras propriedades em conjunto para controlar o comportamento da imagem:

- background-repeat: repete ou não a imagem (no-repeat, repeat-x, repeat-y).
- background-position: define a posição inicial da imagem (top, center, bottom, left, right).
- background-size: ajusta o tamanho (cover, contain, ou valores específicos).

Exemplo combinado:

```
div.banner {  
  
  background-image: url('banner.jpg');  
  
  background-repeat: no-repeat;  
  
  background-size: cover;  
  
  background-position: center;  
  
}
```

O uso eficaz do plano de fundo contribui para a identidade visual da página e para a experiência do usuário, devendo ser escolhido com cuidado para garantir contraste adequado e legibilidade.

4. Padding, margin e bordas básicas

O modelo de caixa do CSS (box model) organiza cada elemento HTML como uma "caixa" que pode ter espaçamentos internos e externos, além de bordas visíveis. Três propriedades controlam esse comportamento: padding, margin e border.

4.1 Padding (espaçamento interno)

padding controla o espaço **dentro do elemento**, entre seu conteúdo e sua borda.

Exemplo:

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

É possível definir valores diferentes para cada lado:

```
padding: 10px 20px 15px 5px; /* superior, direita, inferior, esquerda */
```

4.2 Margin (espaçamento externo)

margin define o espaço **fora da borda do elemento**, separando-o dos elementos vizinhos.

Exemplo:

```
section {  
    margin: 40px 0;
```

```
}
```

O uso estratégico de margin é essencial para o espaçamento entre seções da página e para a centralização de elementos (margin: 0 auto).

4.3 Border (bordas)

A propriedade border adiciona uma borda visível ao redor do elemento. Ela pode ser personalizada em termos de **largura, estilo e cor**.

Exemplo:

```
img {  
  border: 2px solid black;  
}
```

Estilos possíveis incluem solid, dashed, dotted, double, entre outros.

Para bordas arredondadas, usa-se border-radius.

Exemplo:

```
button {  
  border-radius: 10px;  
}
```

5. Considerações finais

O domínio das propriedades de **estilização de texto**, **plano de fundo**, **espaçamentos** e **bordas** é indispensável para qualquer desenvolvedor ou designer web. Esses recursos são os pilares da construção visual de interfaces e permitem criar páginas agradáveis, bem organizadas e responsivas.

Combinando fontes adequadas, tamanhos equilibrados, planos de fundo consistentes e uso inteligente do box model, é possível criar experiências visuais que reforçam a usabilidade e transmitem profissionalismo. O CSS, ao permitir o controle preciso desses aspectos, se torna uma das ferramentas mais poderosas para o desenvolvimento web moderno.

The logo is a large, light blue hexagon with a 3D effect, composed of several smaller, slightly offset hexagonal facets. In the center of this hexagon, the text "Portal" is written in a large, grey, sans-serif font. Below it, the word "IDEA" is written in a larger, bold, grey, sans-serif font. At the bottom, ".com.br" is written in a smaller, grey, sans-serif font.

Portal
IDEA
.com.br

Referências Bibliográficas

- DUCKETT, Jon. *HTML and CSS: Design and Build Websites*. Wiley, 2011.
- FREEMAN, Elisabeth; ROBSON, Eric. *Use a Cabeça! HTML e CSS*. Alta Books, 2015.
- MDN Web Docs – Mozilla Developer Network. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>
- W3Schools. *CSS Styling Tutorial*. Disponível em: <https://www.w3schools.com/css>
- CASTRO, Elizabeth; HYSLOP, Bruce. *HTML5 e CSS3: Guia Prático e Visual*. Alta Books, 2013.

Portal
IDEA
.com.br