



Estilizando

Introdução

Nesta aula sobre estilização, exploraremos técnicas avançadas para aprimorar o design de páginas web, focando especificamente na aplicação de gradientes. Gradientes são uma poderosa ferramenta que permite criar transições suaves de cores, adicionando profundidade e dinamismo aos elementos da sua página. Ao longo da aula, abordaremos como aplicar gradientes em diferentes direções e formas, como linear, radial e cônico, utilizando a propriedade 'background' do CSS. Veremos como essas técnicas podem ser combinadas para criar efeitos visuais impactantes e esteticamente agradáveis.

Bordas

Nesta primeira parte, vamos explorar como trabalhar com bordas em HTML e CSS. As bordas são essenciais na estilização de páginas e aplicações, e aprenderemos as diversas formas de personalizá-las para melhorar o design de nossos layouts.

Primeiramente, é importante entender o que são bordas. Agora, veja abaixo um exemplo prático de como aplicar diferentes estilos de bordas utilizando HTML e CSS:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplos de Bordas</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="borda-solida">Borda Sólida</div>
  <div class="borda-pontilhada">Borda Pontilhada</div>
  <div class="borda-tracejada">Borda Tracejada</div>
  <div class="borda-dupla">Borda Dupla</div>
</body>
</html>
```

CSS (styles.css):

```
div {
  margin: 20px;
  padding: 15px;
  width: 200px;
  text-align: center;
}

.borda-solida {
  border: 2px solid green;
}

.borda-pontilhada {
  border: 2px dotted blue;
}
```

```
.borda-tracejada {
  border: 2px dashed red;
}

.borda-dupla {
  border: 4px double purple;
}
```

Nesse exemplo, diferentes estilos de bordas (sólida, pontilhada, tracejada e dupla) são aplicados aos elementos 'div'. Isso permite visualizar como cada estilo de borda pode ser usado para criar efeitos visuais distintos em uma página.

Em termos simples, uma borda é uma linha que contorna um elemento, como um texto, uma imagem ou uma caixa de conteúdo. No CSS, há várias maneiras de estilizar essas bordas. Podemos aplicar diferentes estilos, cores e tamanhos às bordas dos elementos para criar efeitos visuais distintos.

Entre os estilos de borda disponíveis, temos opções como borda sólida, pontilhada, tracejada e dupla. Cada um desses estilos pode ser combinado com diversas cores para se adequar ao design desejado. Além disso, podemos usar bordas com efeitos especiais, como relevo, crista, embutido ou saliente. Para bordas mais complexas, podemos até mesmo misturar diferentes estilos e aplicar imagens como bordas.

Para aplicar essas bordas, começamos definindo a estrutura no HTML. Em seguida, utilizamos o CSS para estilizar as bordas de acordo com o que foi definido. No HTML, criamos elementos de exemplo, como divs, para aplicar diferentes estilos de bordas. No CSS, usamos propriedades como 'border-style', 'border-color' e 'border-width' para definir o estilo, a cor e a espessura das bordas.

Vamos agora para um exemplo prático. No código HTML, definimos uma estrutura básica com várias divs, cada uma representando um estilo diferente de borda. No CSS, estilizamos cada uma dessas divs aplicando os diferentes estilos de borda e ajustando as cores e espessuras conforme necessário. O resultado é uma visualização clara de como cada estilo de borda se comporta.

Por exemplo, podemos ter uma borda sólida verde em uma div, uma borda pontilhada azul em outra e assim por diante. Ao remover o arquivo CSS,

veremos que as divs ainda estão presentes, mas sem os estilos aplicados, evidenciando a importância do CSS para a estilização visual.

Sombras

Na segunda parte da nossa aula, vamos explorar como aplicar e personalizar sombras em elementos HTML usando CSS. Veja abaixo um exemplo prático de como aplicar diferentes tipos de sombras utilizando HTML e CSS:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplos de Sombras</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="sombra-simples">Sombra Simples</div>
  <div class="sombra-difusa">Sombra Difusa</div>
  <div class="sombra-interna">Sombra Interna</div>
  <div class="sombra-colorida">Sombra Colorida</div>
</body>
</html>
```

CSS (styles.css):

```
div {  
  margin: 20px;  
  padding: 20px;  
  width: 200px;  
  background-color: white;  
  text-align: center;  
  font-family: Arial, sans-serif;  
}  
  
.sombra-simples {  
  box-shadow: 5px 5px 10px #888888;  
}
```

```
.sombra-difusa {  
  box-shadow: 10px 10px 20px #aaaaaa;  
}  
  
.sombra-interna {  
  box-shadow: inset 5px 5px 10px #666666;  
}  
  
.sombra-colorida {  
  box-shadow: 5px 5px 10px rgba(0, 0, 255, 0.5);  
}
```

O exemplo acima aplica diferentes tipos de sombras ('box-shadow') a elementos 'div'. As sombras simples, difusas, internas e coloridas mostram como a propriedade 'box-shadow' pode ser usada para criar efeitos visuais variados, adicionando profundidade e realce aos elementos da página.

Assim como discutimos anteriormente sobre bordas, as sombras são cruciais para melhorar a aparência e a profundidade dos nossos layouts. Com sombras bem aplicadas, é possível criar um efeito mais sofisticado e profissional nas páginas e aplicações.

No CSS, podemos aplicar sombras a diferentes elementos, como caixas de conteúdo e textos, para adicionar profundidade e realçar o visual. As sombras podem ser simples, expandidas, internas, desfocadas, coloridas

ou múltiplas. Vamos ver cada um desses tipos de sombras em detalhes e como aplicá-las no código.

Para ilustrar esses conceitos, começamos definindo a estrutura HTML, que é semelhante à usada para bordas. Criamos uma div principal, chamada de 'container', e para cada tipo de sombra, adicionamos divs específicas. Por exemplo, temos divs para sombras simples, expandidas, internas, desfocadas, coloridas e múltiplas. Cada uma dessas divs possui um parágrafo interno para demonstrar o efeito da sombra.

No CSS, usamos a propriedade 'box-shadow' para adicionar e configurar as sombras. Esta propriedade permite especificar o deslocamento da sombra, o desfoque, a cor e o tamanho. Por exemplo, uma sombra simples pode ser criada com valores de deslocamento e desfoque modestos, enquanto uma sombra expandida utiliza valores maiores para criar um efeito mais pronunciado. Podemos ajustar a cor da sombra usando valores RGB ou nomes de cores para personalizar ainda mais o efeito.

Ao alterar os valores da propriedade 'box-shadow', podemos observar como diferentes combinações criam efeitos variados. Por exemplo, ao aumentar o desfoque, a sombra se torna mais suave e espalhada. Alterar a cor permite experimentar com diferentes tonalidades e intensidades. Para cada tipo de sombra, as mudanças nos valores produzem efeitos visuais distintos, facilitando a personalização dos elementos em uma página.

Para resumir, a aplicação de sombras é uma técnica poderosa para enriquecer o design de páginas web. Com as propriedades corretas e experimentando com diferentes valores, você pode criar efeitos visuais atraentes e funcionais.

Opacidade

Agora, vamos explorar o conceito de opacidade e como aplicá-la aos elementos em HTML e CSS. Veja um exemplo prático de como aplicar diferentes níveis de opacidade utilizando HTML e CSS:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplos de Opacidade</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="opacidade-total">100% Opaco</div>
  <div class="opacidade-alta">75% Opaco</div>
  <div class="opacidade-media">50% Opaco</div>
  <div class="opacidade-baixa">25% Opaco</div>
  <div class="opacidade-minima">10% Opaco</div>
</body>
</html>
```

CSS (styles.css):

```
div {
  margin: 20px;
  padding: 20px;
  width: 200px;
  background-color: #00aaff;
  color: white;
  text-align: center;
  font-family: Arial, sans-serif;
}

.opacidade-total {
  opacity: 1;
}
```

```
.opacidade-alta {  
  opacity: 0.75;  
}  
  
.opacidade-media {  
  opacity: 0.5;  
}  
  
.opacidade-baixa {  
  opacity: 0.25;  
}  
  
.opacidade-minima {  
  opacity: 0.1;  
}
```

Esse exemplo demonstra como aplicar diferentes níveis de opacidade a elementos 'div'. A propriedade 'opacity' é ajustada para mostrar como a transparência afeta a visibilidade dos elementos na página, com valores variando de 100% (totalmente opaco) a 10% (quase transparente).

A opacidade controla o nível de visibilidade de um elemento. Em termos simples, ela determina o quanto um elemento é visível ou transparente. Por exemplo, um elemento com 100% de opacidade é completamente visível, enquanto um elemento com 10% de opacidade é quase invisível. Essa característica é útil para criar efeitos visuais interessantes e para destacar ou suavizar elementos em uma página.

No CSS, a propriedade 'opacity' é usada para definir a transparência. Os valores vão de 0 (totalmente transparente) a 1 (totalmente opaco). Você pode aplicar diferentes valores de opacidade a elementos como bordas, caixas de conteúdo e textos. Por exemplo, ao definir 'opacity: 0.75', o elemento será 75% opaco e 25% transparente. Esses valores podem ser ajustados conforme necessário para criar o efeito desejado.

Para ilustrar a aplicação da opacidade, vamos criar um exemplo prático no código. Em uma estrutura HTML básica, definimos uma div principal

chamada container, que agrupa outros elementos. Dentro do container, criamos várias divs, cada uma representando diferentes níveis de opacidade: 100%, 75%, 50%, 25% e 10%. Cada uma dessas divs pode conter texto ou outros conteúdos, e sua opacidade é ajustada usando a propriedade CSS 'opacity'.

No código CSS, definimos a propriedade 'opacity' para cada uma dessas divs. Por exemplo, para a div com opacidade de 100%, usamos 'opacity: 1', e para a div com opacidade de 25%, usamos 'opacity: 0.25'. Esses ajustes permitem visualizar como a transparência afeta a aparência dos elementos na página.

Assim, ao trabalhar com opacidade, você pode criar efeitos visuais que ajudam a destacar ou suavizar elementos em seu design. A prática com diferentes valores e combinações permitirá que você ajuste a transparência de maneira eficaz, melhorando a estética e a funcionalidade das suas páginas web.

Gradientes

Por último, abordaremos os gradientes, uma técnica poderosa para criar transições suaves de cor em uma página web. Eles permitem que as cores se misturem de maneira fluida, criando efeitos visuais atraentes e dinâmicos. Podemos aplicar gradientes em várias direções, como da esquerda para a direita, do centro para as bordas, entre outros. Observe a seguir um exemplo prático de como aplicar diferentes tipos de gradientes utilizando HTML e CSS:

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplos de Gradientes</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="gradiente-linear">Gradiente Linear</div>
  <div class="gradiente-radial">Gradiente Radial</div>
  <div class="gradiente-conico">Gradiente Cônico</div>
</body>
</html>

```

CSS (styles.css):

```

div {
  margin: 20px;
  padding: 50px;
  color: white;
  font-family: Arial, sans-serif;
  text-align: center;
  border-radius: 8px;
}

.gradiente-linear {
  background: linear-gradient(90deg, #ff7e5f, #feb47b);
}

```

```

.gradiente-radial {
  background: radial-gradient(circle, #43cea2, #185a9d);
}

.gradiente-conico {
  background: conic-gradient(from 90deg, #ff6a00, #ee0979);
}

```

O exemplo acima aplica diferentes tipos de gradientes ('linear-gradient', 'radial-gradient', 'conic-gradient') a elementos 'div'. Os gradientes criam

transições suaves de cor em diferentes direções e formas, proporcionando um visual dinâmico e atraente aos elementos da página.

Os gradientes são uma extensão das técnicas que já exploramos, como bordas, sombras e opacidade. A propriedade CSS usada para gradientes é 'background', que pode ser configurada com valores como 'linear-gradient', 'radial-gradient', 'conic-gradient', e 'repeating-linear-gradient', entre outros. Esses valores definem como as cores se misturam e se distribuem ao longo de um elemento.

Para ilustrar a aplicação dos gradientes, vamos trabalhar com alguns exemplos práticos. No HTML, criamos uma estrutura simples com várias 'divs', cada uma representando um tipo de gradiente: linear, radial, cônico, diagonal e até gradientes aplicados ao texto. A 'div' principal, com a classe 'container', agrupa todas essas 'divs', enquanto cada uma delas mostra um estilo diferente de gradiente.

No CSS, configuramos os gradientes usando a propriedade 'background'. Por exemplo, um gradiente linear pode ser definido com 'background: linear-gradient(90deg, cor1, cor2)', onde '90deg' indica a direção do gradiente e 'cor1 e cor2' são as cores que irão se misturar. Um gradiente radial, configurado com 'background: radial-gradient(cor1, cor2)', cria um efeito circular de transição de cores, começando com cor1 no centro e se esvanecendo para 'cor2' nas bordas. Já o gradiente cônico, com background: 'conic-gradient(cor1, cor2)', apresenta uma transição em forma de círculo a partir do centro.

Além disso, podemos aplicar gradientes ao texto usando a propriedade 'background-clip: text' e outras propriedades para garantir que o gradiente seja visível apenas no texto. Essa técnica é útil para criar efeitos visuais únicos e impactantes.

Em conclusão, a estilização de páginas web é enriquecida significativamente com a aplicação de técnicas como bordas, sombras,

opacidade e gradientes. Cada uma dessas técnicas oferece um conjunto único de ferramentas para aprimorar a aparência e a funcionalidade dos elementos em seus projetos. Bordas e sombras ajudam a definir e destacar elementos, enquanto a opacidade permite ajustes na visibilidade e transparência. Os gradientes, por sua vez, proporcionam transições de cor suaves que podem dinamizar o design. Com a prática e experimentação dessas técnicas, você poderá criar layouts visualmente atraentes e altamente personalizáveis, melhorando a experiência do usuário em suas aplicações web.

Conteúdo Bônus

Para se aprofundar no tema desta aula, sugiro que assista ao vídeo intitulado “3 usos simples e práticos de CSS gradient (efeito degradê)”, que está disponível no canal Mayk Brito no YouTube. Nesse vídeo, você verá 3 usos de efeito degradê com CSS gradient, como aplicar em fundos, em textos e em bordas de elementos.

Referência Bibliográfica

BONATTI, D. **Desenvolvimento de Sites Dinâmicos com Dreamweaver CC**. Brasport: 2013.

BONATTI, D. **Desenvolvimento de Jogos em HTML5**. Brasport: 2014.

FLATSCHART, F. HTML 5 - **Embarque Imediato**. Brasport: 2011.

JOÃO, B. do N. (Org.). **Informática aplicada**. 2.ed. Pearson: 2019.

MARINHO, A. L.; CRUZ, J. L. da. **Desenvolvimento de aplicações para Internet**. 2.ed. Pearson: 2020

NEVES, M. C. B. de A. **Sites de Alta Performance**. Contentus: 2020

SOUSA, R. F. M. CANVAS HTML 5 - **Composição gráfica e interatividade na web**. Brasport: 2018.

TANENBAUM, A. S.; FEAMSTER, N.; WETHERALL, D. J. **Redes de computadores**. 6.ed. Pearson: 2021.

Atividade Prática 7 - Estilizando

Título da Prática: Estilizando com CSS e HTML

Objetivos:

- Objetivo 1 – Aprendendo a criar estilos com CSS.
- Objetivo 2 – Integrando estilos com CSS e HTML.
- Objetivo 3 – Aplicando estilos de forma prática com CSS.

Materiais, Métodos e Ferramentas:

- Ferramenta 1 – IDE.
- Método 1 – Atividade Prática.
- Materiais 1 – Acesso ao material complementar de aula.

Atividade Prática

CSS (Cascading Style Sheets) é usado para estilizar elementos HTML, aplicando cores, fontes, layouts e outros efeitos visuais. Nesta atividade, você irá criar uma página HTML e aplicar diferentes estilos com CSS para melhorar a aparência da página.

Tarefa:

Crie uma página HTML com o seguinte conteúdo:

Um **título principal** (h1) com o texto: “Estilizando com CSS”.

Um **parágrafo** com o texto: “CSS facilita a personalização de páginas web”.

Um **botão** com o texto: “Clique aqui”.

Estilize a página da seguinte maneira:

O título deve ter uma cor de fundo **vermelha** e texto **branco**.

O parágrafo deve ter **font-size de 18px** e o texto justificado.

O botão deve ter um **fundo azul**, texto **branco** e deve mudar de cor quando o mouse passar por cima.

Publique o arquivo no navegador para verificar o resultado.

Passos a seguir:

1 - Crie um arquivo HTML com a extensão **.html** (por exemplo, pagina_estilizada.html).

2 - Adicione a estrutura básica de um documento HTML.

3 - Insira o título, parágrafo e botão no HTML.

4 - Estilize o título, parágrafo e botão usando CSS.

5 - Abra o arquivo no navegador para visualizar o resultado.

Gabarito Atividade Prática

Passo 1:

Aqui está o código HTML e CSS para estilizar os elementos:

html

Copiar código

```
<!DOCTYPE html>
```

```
<html lang="pt-BR">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Estilizando com CSS</title>
```

```
  <style>
```

```
    /* Estilo do título /
```

```
    h1 {
```

```
      background-color: red;
```

color: white;

padding: 10px;

text-align: center;

}

/Estilo do parágrafo /

p {

font-size: 18px;

text-align: justify;

margin: 20px;

}

/Estilo do botão /

button {

background-color: blue;

color: white;

border: none;

padding: 10px 20px;

font-size: 16px;

cursor: pointer;

}

/Efeito hover no botão */

```
button:hover {
```

```
    background-color: darkblue;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<!-- Título -->
```

```
<h1>Estilizando com CSS</h1>
```

```
<!-- Parágrafo -->
```

```
<p>CSS facilita a personalização de páginas web.</p>
```

```
<!-- Botão -->
```

```
<button>Clique aqui</button>
```

```
</body>
```

```
</html>
```

Explicação:

Estrutura HTML:

O título é um elemento <h1> que contém o texto principal.

O parágrafo está dentro de uma tag <p>.

O botão está dentro da tag <button>.

Estilização CSS:

Título (h1): O fundo do título é vermelho, o texto é branco, e ele é centralizado com text-align: center.

Parágrafo §: O tamanho da fonte é 18px, o texto é justificado, e um espaçamento de 20px foi adicionado ao redor.

Botão (button): O botão tem um fundo azul, texto branco, e ao passar o mouse (hover) sobre o botão, o fundo muda para um tom de azul mais escuro.

Passo 2:

Salve o arquivo com a extensão .html (por exemplo, pagina_estilizada.html).

Passo 3:

Abra o arquivo no navegador para visualizar a página estilizada. Se você seguiu os passos corretamente, o título estará com fundo vermelho, o parágrafo justificado e o botão terá a interação de mudança de cor ao passar o mouse.

Erros Comuns:

Esquecer de aplicar o estilo ao botão: Certifique-se de que a regra button:hover está corretamente definida para mudar a cor de fundo do botão quando o mouse passa por cima.

Não usar `text-align: justify`; no parágrafo: Isso pode fazer com que o texto não fique uniformemente distribuído nas laterais.

Não aplicar a cor de fundo no título (`h1`): Verifique se `background-color` e `color` estão definidos corretamente.

Essa atividade ensina como aplicar estilos básicos, como cores, alinhamento, tamanhos de fonte e interações de hover, proporcionando uma introdução ao uso do CSS para estilizar elementos em uma página web.

[Ir para exercício](#)