APOSTILA DE REVISÃO V1

HTML 5 & CSS 3

Módulo 3: Fundamentos de CSS 3

FORMULA WEB PRO-MAX

Introdução ao Módulo de Fundamentos do CSS

Bem-vindo ao emocionante mundo dos **Fundamentos do CSS 3** Moderno! Neste módulo, iremos embarcar em uma jornada que elevará as suas habilidades de desenvolvimento web e fornecerá conhecimentos e técnicas essenciais para criar interfaces de usuário deslumbrantes e responsivas.

Vamos explorar os avanços mais significativos do CSS 3 para que você seja capaz de criar layouts que se adaptam perfeitamente a diferentes tamanhos de tela e dispositivos. Abordaremos os conceitos de Flexbox e Grid, liberando todo o seu potencial para alcançar designs sofisticados e responsivos com facilidade.

Mas isso não é tudo! Você também aprenderá sobre transições e animações CSS, dando vida às suas páginas web com efeitos cativantes que encantarão os visitantes das suas páginas. Além disso, abordaremos o tópico de variáveis CSS, permitindo que você escreva um código mais limpo e de fácil manutenção, aproveitando o poder de valores reutilizáveis.

Ao final deste módulo, você estará apto não apenas a criar interfaces de usuário impressionantes, mas também a otimizar o desempenho do seu desenvolvimento e estar atualizado com as melhores práticas de design web.

Portanto, prepare-se para uma jornada web pro max enquanto desvendamos o mundo dos Fundamentos do CSS 3 Moderno. Vamos embarcar juntos nesta experiência de aprendizado transformadora e elevar suas habilidades de desenvolvimento web a novas alturas!

Implementando CSS

Existem várias formas de implementar CSS em um documento HTML, e cada abordagem tem suas vantagens e desvantagens. Vou descrever as principais formas de implementação de CSS:

CSS Interno (Embedded CSS):

Nessa abordagem, o CSS é incorporado diretamente no documento HTML usando a tag <style>. O código CSS fica dentro das tags <style> na seção <head> do documento HTML. Essa forma de implementação é útil para aplicar estilos específicos apenas a esse documento HTML.

CSS Externo (External CSS):

Nessa abordagem, o CSS é armazenado em um arquivo separado com extensão .css o arquivo CSS externo é vinculado ao documento HTML usando a tag <link>. Essa forma de implementação é recomendada para projetos maiores, pois permite a reutilização do mesmo arquivo CSS em várias páginas.

CSS Inline (Inline CSS):

Nessa abordagem, o CSS é aplicado diretamente a um elemento HTML específico usando o atributo style. O código CSS fica dentro do atributo style do elemento. Essa forma de implementação é útil para aplicar estilos a um único elemento de forma rápida, mas não é recomendada para estilização geral de uma página inteira.

Seletores Básicos em CSS 3

Os seletores CSS (Cascading Style Sheets) são uma parte essencial do CSS, permitindo que você selecione e estilize elementos HTML específicos em uma página da web. Entender os seletores CSS é crucial para personalizar a aparência e o layout das suas páginas da web. Aqui está uma visão geral dos seletores CSS mais comumente utilizados:

Seletor Universal (*): o asterisco (*) seleciona todos os elementos na página. **Exemplo:** * { color: blue; } tornará a cor do texto de todos os elementos na página azul.

Seletor de Tipo de Tag: o seletor de tipo direciona elementos pelo nome da tag HTML. **Exemplo:** p { font-size: 16px; } definirá o tamanho da fonte de todos os parágrafos para 16 pixels.

Seletor de Classe (.): o seletor de classe direciona elementos com um atributo de classe específico. **Exemplo:** .minha-classe { color: red; } tornará a cor do texto de elementos com class="minha-classe" vermelha.

Seletor de ID (#): o seletor de ID direciona um elemento específico com um atributo de ID único. **Exemplo:** #meu-id { background-color: yellow; } definirá a cor de fundo do elemento com id="meu-id" como amarela.

É importante usar os seletores de forma eficaz e eficiente para manter o código CSS organizado e fácil de manter. Combinar vários seletores e entender a especificidade dos seletores ajudará você a criar regras CSS poderosas e precisas para estilizar suas páginas da web.

Introdução a Dev Tools

As ferramentas de desenvolvedor (Dev Tools) são recursos integrados nos navegadores modernos que permitem aos desenvolvedores inspecionar, depurar e editar o código de uma página da web em tempo real. Elas fornecem um conjunto de utilitários poderosos para desenvolvedores web aprimorarem e otimizarem seus sites ou aplicativos. A Dev Tools é uma ferramenta indispensável para o desenvolvimento front-end, especialmente quando se trata de trabalhar com CSS. Aqui estão algumas das principais funcionalidades das Dev Tools relacionadas ao CSS:

1. **Inspecionar Elementos (Inspect Element):** Com as Dev Tools, você pode inspecionar qualquer elemento na página simplesmente clicando com o botão direito do mouse sobre ele e selecionando "Inspecionar" ou usando atalhos do teclado (por exemplo, Ctrl+Shift+C). Ao inspecionar um elemento, você pode ver a estrutura HTML e as regras CSS aplicadas ao elemento, incluindo o box model, cores, tamanhos, fontes e outras propriedades.
2. **Edição ao Vivo (Live Editing):** As Dev Tools permitem fazer edições ao vivo no código CSS e ver as alterações instantaneamente sem precisar recarregar a página. Você pode editar propriedades CSS, adicionar novas regras ou ajustar valores para refinar o layout ou estilo da página.
3. **Seletor de Elementos (Element Selector):** Ao usar a ferramenta de seleção de elementos, você pode clicar em um elemento na página para ver imediatamente os seletores CSS aplicados a ele. Isso ajuda a identificar rapidamente as regras de estilo que afetam um elemento específico.
4. **Inspeção de Estilos Computados (Computed Styles):** A guia "Styles" nas Dev Tools exibe os estilos computados de um elemento, mostrando todas as propriedades CSS aplicadas a ele, incluindo as herdadas e as que são resultado de regras de cascata.

5. **Visualização de Caixas (Box Model Visualization):** As Dev Tools permitem visualizar as caixas dos elementos (conteúdo, padding, border e margin) através de uma representação gráfica, o que ajuda a entender o layout e o espaçamento dos elementos na página.
6. **Media Query Inspection:**As Dev Tools mostram como as regras de mídia (media queries) estão afetando o layout e os estilos em diferentes tamanhos de tela ou dispositivos, facilitando o desenvolvimento responsivo.
7. **Desativar Estilos (Disable Styles):** Você pode temporariamente desativar estilos específicos ou até mesmo todos os estilos de uma página para entender como eles afetam o layout e a aparência do site.
8. **Exibição de Grade (Grid Overlay):** Para layouts criados com CSS Grid ou Flexbox, as Dev Tools oferecem sobreposições visuais para ajudar a ver as áreas e os itens do grid, facilitando o ajuste e a solução de problemas.

As Dev Tools são uma ferramenta essencial para desenvolvedores front-end trabalharem eficientemente com CSS. Elas permitem uma abordagem mais iterativa e interativa para o desenvolvimento, permitindo que você experimente, teste e refine o código CSS em tempo real, resultando em um design mais eficaz e atraente para sua página da web.

Fontes em CSS 3

Fontes: permitem que você controle o tipo de letra, tamanho, peso, estilo e outras propriedades relacionadas ao texto.

Visão geral completa sobre fontes em CSS:

font-family (família): usado para especificar as font a ser usada, se a primeira fonte da lista não estiver disponível, o navegador tentará a próxima. **Ex:** p { font-family: Arial, sans-serif; }

font-size (tamanho): determina a altura dos caracteres do texto em pixels (px), pontos (pt), unidades em (em), percentagem (%) etc.

Ex: p { font-size: 16px; }

font-weight (espessura): define a espessura ou negrito dos caracteres do texto.Valores comuns são normal, bold, lighter, bolder e valores numéricos (100, 200, ..., 900). **Ex:** p { font-weight: bold; }

font-style (estilo): especifica se o texto deve ser renderizado em itálico ou oblíquo. **Ex:** p { font-style: italic; }

text-transform (transformação): altera a capitalização do texto. Valores incluem uppercase (maiúsculas), lowercase (minúsculas), capitalize (inicial maiúscula) e none (nenhuma transformação).

Ex: p { text-transform: uppercase; }

text-decoration (decoração): adiciona efeitos visuais ao texto, como sublinhado ou sobrelinha. Valores incluem underline (sublinhado), overline (sobrelinha), line-through (riscado), none (nenhuma decoração) e blink (sim, texto piscante!). **Ex:** p { text-decoration: underline; }

text-shadow (sombra): adiciona um efeito de sombra ao texto.

Ex: p { text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); } criará uma sombra 2 pixels à direita e 2 pixels abaixo do texto, com um raio de desfoque de 4 pixels e opacidade de 50%.

Fontes Seguras para a Web (Web Safe Fonts): são fontes amplamente disponíveis em diferentes sistemas operacionais e navegadores sem a necessidade de arquivos de fonte adicionais. Ex: Arial, Helvetica, Times New Roman, Georgia, etc.

Pilha de Fontes (Font Stacks): referem-se à especificação de várias fontes de fallback para aumentar a probabilidade de encontrar uma fonte adequada em diferentes dispositivos. Ex: font-family: 'Open Sans', 'Helvetica Neue', Arial, sans-serif;

Tipos de Cores em CSS 3

Cores: desempenham um papel essencial na definição da aparência visual dos elementos em uma página da web. O CSS suporta vários formatos de cores e oferece uma ampla variedade de opções para escolher.

Diferente tipos de cores em css:

Palavras chaves: red, blue, yellow, green, gold, darkblue, etc.

HEX e HEXA: #ff0000; #ff000054

RGB e RGBA: rgb(255, 0, 0); rgba(255, 0, 0, 0.5)

HSL e HSLA: hsl(0, 100%, 50%); hsla(0, 100%, 50%, 0.5)

Background e Bordas

Em CSS, o background (fundo) e as borders (bordas) são propriedades essenciais para controlar a aparência visual dos elementos em uma página da web. Vamos explorar cada uma delas:

Background (Fundo):

A propriedade background permite definir o plano de fundo de um elemento HTML. É possível definir cores, imagens e outros efeitos de fundo.

Exemplo: background-color: #f0f0f0; define o fundo de um elemento com uma cor cinza claro.

Além disso, há outras propriedades relacionadas ao fundo:

- **background-color:** Define a cor de fundo do elemento.
- **background-image:** Define uma imagem de fundo. Pode ser uma imagem local ou uma URL externa.
- **background-repeat:** Especifica como a imagem de fundo deve se repetir ou não.
- **background-position:** Controla a posição da imagem de fundo.
- **background-size:** Define o tamanho da imagem de fundo.
- **background-attachment:** Determina se a imagem de fundo rola junto com o conteúdo ou permanece fixa à medida que o usuário rola a página.

Borders (Bordas):

A propriedade border permite definir bordas ao redor de um elemento HTML.

Exemplo: border: 1px solid black; cria uma borda sólida de 1 pixel de espessura e cor preta.

Existem várias propriedades relacionadas às bordas:

- **border-width:** Define a espessura da borda.
- **border-style:** Especifica o estilo da borda, como sólido, tracejado, pontilhado, entre outros.
- **border-color:** Define a cor da borda.
- **border-radius:** Cria bordas arredondadas, permitindo controlar o raio dos cantos.
- **border-image:** Permite usar uma imagem para estilizar a borda.

Além disso, é possível especificar as propriedades das bordas individualmente:

- **border-top:** Define as propriedades da borda superior.
- **border-right:** Define as propriedades da borda direita.
- **border-bottom:** Define as propriedades da borda inferior.
- **border-left:** Define as propriedades da borda esquerda.

Por exemplo:

- border-top: 1px solid red;
- border-right: 2px dashed blue;
- border-bottom: 3px dotted green;
- border-left: 4px double orange;

Box Model (Modelo de Caixa), Margin (Margem) e Padding (Preenchimento)

O Box Model é um conceito fundamental do CSS que descreve como os elementos HTML são renderizados como caixas retangulares. Cada elemento é tratado como uma caixa, e essa caixa é composta por quatro áreas principais: conteúdo, padding, border e margin. Essas áreas determinam o tamanho total do elemento na página e afetam seu posicionamento e espaçamento em relação a outros elementos. Aqui está uma explicação detalhada de cada parte do Box Model:

Conteúdo (Content):

A área de conteúdo é a parte interna da caixa e abriga o conteúdo real do elemento, como texto, imagens ou outros elementos HTML. A largura e a altura do conteúdo são definidas usando as propriedades width e height.

Padding:

O padding é uma área transparente que circunda o conteúdo dentro da caixa. Ele é usado para criar espaçamento entre o conteúdo e a borda da caixa. A propriedade padding controla o espaçamento nas quatro direções (cima, direita, baixo e esquerda), e as propriedades individuais como padding-top, padding-right, padding-bottom e padding-left podem ser usadas para especificar espaçamentos diferentes para cada direção.

Border:

A borda é uma linha que envolve o conteúdo e o padding da caixa. Ela pode ser sólida, pontilhada, tracejada ou outro estilo definido com a propriedade

border-style. A espessura da borda é controlada pela propriedade border-width, e a corda borda é definida pela propriedade border-color.

Propriedades como border-top, border-right, border-bottom e border-left podem ser usadas para especificar estilos, larguras e cores diferentes para cada lado da caixa.

Margin:

A margem é uma área transparente que fica fora da borda da caixa e é usada para criar espaçamento entre o elemento e outros elementos na página. A propriedade margin controla o espaçamento nas quatro direções, e as propriedades individuais como margin-top, margin-right, margin-bottom e margin-left podem ser usadas para especificar margens diferentes para cada direção.

O cálculo do tamanho total do elemento é feito adicionando a largura do conteúdo e o padding, juntamente com a espessura da borda, e, se houver, a margem externa. A propriedade box-sizing pode ser usada para controlar como o tamanho total do elemento é calculado. O valor content-box (padrão) inclui apenas o conteúdo na largura total, enquanto o valor border-box inclui o conteúdo, padding e borda na largura total, tornando mais fácil o gerenciamento do layout.

O Box Model é um conceito crítico a ser compreendido para criar layouts precisos e bem espaçados em páginas da web. Ele permite controlar o tamanho e o espaçamento dos elementos, garantindo uma aparência consistente e organizada para o conteúdo do seu site.

Floats (Flutuação) e Alinhamento

Float (Flutuação):

A propriedade float é usada para permitir que um elemento de bloco (como uma imagem, por exemplo) flutue em torno de outro elemento. Quando um elemento é definido com float, ele é removido do fluxo normal do documento e posicionado ao longo do lado especificado de seu elemento pai.

Isso pode ser útil para criar layouts de colunas ou para posicionar elementos ao lado de outro conteúdo, como imagens ao lado do texto.

Exemplo:

```
img {
 float: left; /* A imagem flutua a esquerda
  */margin: 10px;
}
<img src="imagem.jpg" alt="Imagem Flutuante">
<p>Este é um parágrafo de texto que ficará ao lado da imagem flutuante.</p>
```

Clear (Limpar Flutuação):

Quando elementos flutuantes são usados, é comum ocorrerem problemas de layout em elementos subsequentes, que podem ficar posicionados ao lado dos elementos flutuantes. Para evitar isso, a propriedade clear é utilizada. A propriedade clear define se um elemento deve ser posicionado abaixo dos elementos flutuantes ou não.

clear: left faz com que o elemento não fique ao lado de elementos flutuantes a esquerda, e clear: right faz o mesmo para elementos flutuantes a direita.

Exemplo:

```
p {
  clear: both; /* O parágrafo não ficará ao lado de elementos flutuantes */
}
```

Alinhamento (text-align):

A propriedade text-align é usada para controlar o alinhamento horizontal do texto dentro de um elemento. Ela também afeta o alinhamento de outros elementos inline, como imagens e elementos de linha (por exemplo,).

Os valores comuns são left (esquerda), right (direita), center (centro) e justify (justificado).

Exemplo:

```
div {
  text-align: center; /* O texto e outros elementos inline dentro do div serão centralizados */
}
```

Alinhamento Vertical (vertical-align):

A propriedade vertical-align é usada para controlar o alinhamento vertical de elementos inline ou elementos de tabela.

Ela pode ser usada para alinhar elementos verticalmente em relação ao texto circundante ou a outras linhas de texto.

Exemplo:

```
img {
  vertical-align: middle; /* A imagem é alinhada verticalmente no centro da linha */
}
```

Esses conceitos de float e alinhamento são úteis para criar layouts flexíveis e controlar o posicionamento de elementos em uma página da web. Eles fornecem ferramentas poderosas para controlar a apresentação visual do conteúdo e permitir uma experiência de leitura e navegação agradável para os visitantes do site.

Estados de Links e Estilização de Botão

Os "estados de links" referem-se aos diferentes estados visuais que um link pode ter, dependendo da interação do usuário ou do status de navegação. Os estados de links são uma parte essencial do CSS e são amplamente utilizados para melhorar a experiência do usuário e fornecer feedback visual durante a interação com links em uma página da web. Os principais estados de links são:

Link Normal (Não Visitado):

O estado padrão de um link é quando o usuário ainda não o visitou. Por padrão, os links não visitados são exibidos com a cor definida pelo navegador ou uma cor padrão do tema do site (geralmente azul) e são sublinhados.

Link Visitado:

Quando o usuário visita um link, ou seja, quando ele clica e acessa a página vinculada, o link entra no estado "visitado". Por padrão, os links visitados são exibidos com uma cor definida pelo navegador (geralmente roxo escuro) e ainda podem ser sublinhados.

Link Hover (Passando o Mouse):

O estado de link "hover" é ativado quando o usuário passa o mouse sobre um link, antes de clicar nele. É comum alterar a cor do link ou o estilo visual, como remover o sublinhado, para fornecer feedback visual de que o link é clicável.

Link Active (Clicando):

O estado "active" é ativado quando o usuário está clicando no link, ou seja, o momento em que o link está sendo ativamente pressionado pelo usuário. É comum aplicar temporariamente um estilo visual para indicar que o link está sendo clicado.

Para controlar os estilos de cada estado de link, você pode usar as pseudo-classes do CSS. As pseudo-classes são notações especiais que permitem selecionar elementos com base em seu estado ou posição no DOM. As pseudo-classes relacionadas a estados de links são:

Exemplo de uso de pseudo-classes para estilizar estados de link:

```
a:link { /* Estilizando links não visitados */
  color: blue;
  text-decoration: underline;
}
a:visited { /* Estilizando links visitados */
  color: purple;
}
a:hover { /* Estilizando links quando o mouse está sobre eles */
  color: red;
  text-decoration: none; /* Removendo o sublinhado no hover */
}
a:active { /* Estilizando links quando estão ativamente sendo clicados */
  color: green;
}
```

Utilizar estados de links de forma adequada ajuda a fornecer feedback visual claro para os usuários, melhorando a usabilidade e a experiência de navegação em seu site ou aplicativo web. É uma prática recomendada aplicar estilos consistentes e intuitivos para garantir que os usuários saibam facilmente quais elementos são clicáveis e quais já foram visitados anteriormente.

Displays Inline, Block e Inline-block

No CSS (Cascading Style Sheets), as propriedades display inline, display block e display inline-block são usadas para controlar o layout e comportamento dos elementos HTML. Cada uma dessas propriedades define como o elemento é renderizado dentro do fluxo normal da página. Vamos explorar cada uma delas:

display: inline;

Elementos com display: inline; são renderizados sem quebra de linha, ou seja, eles se comportam como palavras dentro de um parágrafo. Eles fluem ao longo da linha e ocupam apenas o espaço necessário para o conteúdo interno.

Margem superior e inferior, bem como altura e largura, não são aplicadas a elementos com display: inline;.

Exemplos de elementos que são por padrão inline: , <a>, , , <img>.

display: block;

Elementos com display: block; são renderizados como "blocos" independentes que ocupam a largura total do contêiner pai (a menos que uma largura específica seja definida). Isso significa que eles começam em uma nova linha e empurram quaisquer elementos subsequentes para a linha abaixo.

Margens, altura e largura podem ser aplicadas a elementos com display: block;.

Exemplos de elementos que são por padrão block: , <p>, <h1> a <h6>, <ul>, <li>.

display: inline-block;

inline-block é uma combinação de inline e block. Os elementos com display: inline-block; são renderizados como blocos retangulares, mas não forçam uma quebra de linha após eles. Eles fluem ao longo da linha, permitindo que outros elementos sejam colocados na mesma linha, se houver espaço suficiente.

Margens, altura e largura também podem ser aplicadas a elementos com display: inline-block;.

Essa propriedade é útil quando você deseja combinar as características de block com a capacidade de colocar elementos na mesma linha, como um conjunto de botões ou imagens.

Exemplos de elementos que podem ser definidos como inline-block usando CSS: <img>, <button>, .

Exemplo de uso no CSS:

```
span { /* Exemplo com display inline */
  display: inline;
}
div { /* Exemplo com display block */
  display: block;
}
button { /* Exemplo com display inline-block */
  display: inline-block;
}
```

É importante entender essas diferenças ao criar o layout do seu site, pois a escolha do valor display pode afetar o comportamento dos elementos e a aparência geral da página. Lembre-se de que você também pode alterar o valor display usando JavaScript para adaptar o layout conforme necessário em diferentes situações.

Posicionamento (position)

Em CSS, a propriedade position é usada para controlar o posicionamento de elementos HTML dentro do layout da página. Ela define como um elemento é posicionado em relação ao seu contêiner ou ao documento como um todo. Existem vários valores para a propriedade position, cada um com um comportamento diferente:

position: static;

É o valor padrão para todos os elementos. Nesse modo, o elemento é posicionado conforme o fluxo normal do documento, ignorando outras propriedades de posicionamento. As propriedades top, right, bottom e left não têm efeito quando position é static.

position: relative;

Com position: relative;, o elemento é deslocado de sua posição original em relação a si mesmo. Ele ainda ocupa espaço em seu local original no fluxo do documento, mas pode ser ajustado usando as propriedades top, right, bottom e left. Se você definir top: 10px;, por exemplo, o elemento será movido 10 pixels para baixo a partir de sua posição original.

position: absolute;

Com position: absolute;, o elemento é posicionado em relação ao contêiner mais próximo que possua uma propriedade position diferente de static (ou seja, relative, absolute ou fixed). Ao definir position: absolute;, você pode usar top, right, bottom e left para posicioná-lo exatamente onde deseja no contêiner pai.

Elementos com position: absolute; são removidos do fluxo normal do documento e não ocupam espaço em seu local original.

position: fixed;

Com position: fixed;, o elemento é posicionado em relação à janela do navegador, ou seja, ele permanecerá no mesmo lugar mesmo quando a página for rolada. Esse tipo de posicionamento é útil para criar elementos como barras de navegação fixas no topo ou rodapés fixos na parte inferior da página.

position: sticky;

A propriedade position: sticky; é semelhante ao position: relative até que o elemento atinja um determinado ponto na página (normalmente definido por top, right, bottom ou left), a partir desse ponto, ele se comporta como position: fixed, mantendo-se fixo na posição especificada mesmo enquanto a página é rolada.

O elemento fica "preso" na tela até que o usuário role o suficiente para que ele retorne ao fluxo normal do documento.

A escolha do valor correto para a propriedade position é crucial para criar layouts precisos e responsivos em seu site. Certifique-se de entender como cada valor afeta o posicionamento do elemento em relação aos seus contêineres pai e ao documento como um todo.