

Mão na massa

1) Criando e linkando o arquivo CSS

Você vai criar um arquivo CSS chamado `style.css` para estilizar uma página web. Após criar o arquivo, seu objetivo é linkar esse arquivo CSS ao HTML. Para isso, você usará a tag `<link>` dentro do `<head>` do documento HTML, especificando o caminho do arquivo CSS. Este processo é crucial para que os estilos definidos no CSS sejam aplicados à página HTML.

2) Definindo estilos básicos

No arquivo CSS, defina a cor de fundo do corpo da página (`<body>`) para indigo (*indigo*) e a cor do texto para branco (*white*). Salve as alterações e verifique no navegador se o fundo da página está preto e o texto branco.

3) Estilizando com cores e formatos

Você recebeu o desafio de dar vida a uma página web que atualmente tem apenas HTML. Sua tarefa é utilizar CSS para adicionar cores e formatos aos elementos da página, inspirando-se em um design específico fornecido no Figma. Foque em selecionar cores adequadas para os fundos, textos e bordas, além de aplicar estilos aos botões e cabeçalhos para torná-los visualmente atraentes. Você pode usar como referência essa [tabela de nomes de cores](#).

4) Personalizando a página com CSS

Imagine que você está trabalhando em um projeto web e chegou o momento de dar vida à sua página com cores, formas e estilos. Sua tarefa é criar um arquivo CSS para

estilizar uma página HTML simples, a qual contém um cabeçalho, um parágrafo e um rodapé. Use seu conhecimento para definir as cores do texto e do fundo.

5) Aplicando estilos com CSS

Agora, vamos um pouco além. Você precisa criar três arquivos CSS diferentes, cada um com um estilo único, para o mesmo arquivo HTML. Isso demonstrará como o CSS pode drasticamente mudar a aparência de uma página sem alterar o HTML. Crie estilos variados para os elementos da página - um arquivo com um estilo formal e elegante, outro com um estilo colorido e divertido, e um terceiro com um visual minimalista e limpo.

6) Personalizando Links

A última tarefa é estilizar os links na página. Você deverá modificar a aparência dos links para que se destaquem do fundo e do texto. Utilize as propriedades CSS, como `color` para personalizá-los, seguindo as diretrizes do design fornecido no Figma.

Opinião do instrutor

1) Criando e linkando o arquivo CSS

1 - Primeiro, crie o arquivo `style.css` na pasta do projeto.

2 - No documento HTML, dentro do `<head>`, insira a tag `<link>` para conectar o arquivo CSS:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <!-- Outras metainformações -->
  <title>Portfolio</title>
```

```
<link rel="stylesheet" href="style.css">
</head>
<!-- Restante do HTML -->
```

2) Definindo estilos básicos

- No arquivo style.css, defina a cor de fundo do `<body>` para preto e a cor do texto para branco.

Código CSS

```
body {
  background-color: indigo;
  color: white;
}
```

3) Estilizando com cores e formatos

1 - Primeiramente, defina as cores dos elementos principais da página, como o fundo (`background-color`), texto (`color`) e bordas (`border`). Por exemplo:

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

h1 {
  color: #333333;
}
```

2 - Aplique estilos aos botões (`button`) e cabeçalhos (`h1`, `h2`, etc.) para torná-los mais atraentes, usando propriedades como `border-radius`, `padding` e `font-weight`:

```
button {
  background-color: green;
```

```
color: white;
}
```

4) Personalizando a página com CSS

1 - Crie um arquivo chamado `estilo.css`.

2 - No arquivo `estilo.css`, defina o estilo do cabeçalho (`header`) com uma cor de fundo e cor do texto. Exemplo:

```
header {
    background-color: lightblue;
    color: white;
}
```

3 - Estilize o parágrafo (`p`) com um cor de fonte agradável. Por exemplo:

```
p {

    color: darkgray;
}
```

4 - Para o rodapé (`footer`), defina uma cor de fundo distinta:

```
footer {
    background-color: gray;
    color: white;
}
```

5) Aplicando estilos com CSS

Para cada estilo:

1 - **Estilo Formal e Elegante:** Crie `estilo1.css`.

- Use cores sóbrias, como preto, cinza e branco.
- Exemplo:

```
body {
    background-color: white;
```

```
color: black;  
}
```

2 - **Estilo Colorido e Divertido:** Crie `estilo2.css`.

- Utilize cores vibrantes e variações de fontes mais descontraídas.
- Exemplo:

```
body {  
  background-color: yellow;  
  color: blue;  
}
```

3 - **Estilo Minimalista e Limpo:** Crie `estilo3.css`.

- Prefira um esquema de cores minimalista, como branco sobre preto.
- Exemplo:

```
body {  
  background-color: black;  
  color: white;  
}
```

6) Personalizando links

1 - Personalize os links para que se destaquem no design. Por exemplo:

```
a {  
  color: DodgerBlue;  
}
```

2 - Salve as alterações e atualize a página no navegador para ver o resultado.