

Universidade de Pernambuco
Professor: Augusto César Oliveira
Disciplina: Programação IV
Aluno: Gabriel Gomes Barbosa

Respostas da Aula 02 - Introdução ao CSS

Questão 1:

CSS é uma linguagem de estilo usada para descrever a apresentação de um documento escrito em HTML. Seu papel no desenvolvimento web é separar o conteúdo (HTML) da aparência visual (estilo), facilitando a manutenção, reutilização de código e melhorando a acessibilidade e performance.

Questão 2:

JavaScript e frameworks como React são comumente usados com CSS. Essa integração permite criar páginas dinâmicas e interativas, melhorando a usabilidade e possibilitando estilos condicionais e animações, o que enriquece a experiência do usuário.

Questão 3:

Uma regra CSS é composta por: seletor (define o elemento HTML a ser estilizado), propriedades (atributos que serão modificados) e valores (especificações para as propriedades). Exemplo:

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

Questão 4:

O navegador interpreta o CSS com base na cascata, especificidade e ordem de declaração. Ele aplica a regra mais específica ao elemento. Se houver conflitos, a regra com maior especificidade ou a última declarada prevalece.

Questão 5:

Seletor de tipo seleciona elementos pelo nome da tag (ex: `div`), classe usa o atributo `class` (ex: `.container`), e ID usa o atributo `id` (ex: `#menu`). Classes são reutilizáveis em vários elementos, enquanto IDs devem ser únicos por página. Classes são preferidas quando o estilo precisa ser aplicado a múltiplos elementos.

Questão 6:

Pseudo-elementos permitem estilizar partes específicas de um elemento, como o primeiro caractere ou linha. Exemplos:

```
::first-line { font-weight: bold; }
```

```
::first-letter { font-size: 200%; }
```

Questão 7:

O box model representa cada elemento HTML como uma caixa retangular composta por: conteúdo, padding (espaço interno), border (borda) e margin (espaço externo). Esses componentes influenciam o tamanho e o espaçamento entre elementos na página.

Questão 8:

O modelo de caixa permite ajustar margens para separar elementos, padding para criar espaço interno e bordas para destacar seções. Exemplo: aplicar margin-bottom para espaçar parágrafos, ou padding em um botão para torná-lo mais clicável.

Questão 9:

Flexbox é um modelo de layout unidimensional que organiza elementos em linha ou coluna. Principais conceitos incluem os eixos principal e transversal, alinhamento (justify-content, align-items) e distribuição de espaço. Ele facilita a criação de layouts flexíveis e responsivos com menos código.

Questão 10:

Flexbox é ideal para layouts lineares (um eixo), enquanto Grid é melhor para layouts bidimensionais (linhas e colunas). Grid é preferido para estruturas mais complexas, enquanto Flexbox é mais simples para alinhamentos e distribuições em linha ou coluna.

Questão 11:

Propriedades: font-family, font-size, color. Usar uma fonte legível, tamanho adequado e cores com bom contraste melhora a leitura e aparência do texto.

Questão 12:

Essas propriedades ajustam o espaçamento entre linhas e letras, melhorando a legibilidade e tornando o texto visualmente mais agradável.

Questão 13:

A propriedade `color` define a cor do texto, e `background-color` define a cor de fundo de um elemento. Exemplo:

```
h1 { color: white; background-color: blue; }
```

Questão 14:

CSS permite gradientes com `linear-gradient` ou `radial-gradient`. Gradientes criam transições suaves entre cores, adicionando profundidade e estética ao design.

Questão 15:

Pseudoclasses definem estilos para estados específicos de elementos. Exemplos:

`:hover` (quando o mouse passa sobre o elemento)

`:active` (ao clicar)

`:visited` (link já visitado)

`:focus` (quando o elemento recebe foco, como um input).

Questão 16:

`:nth-child()`, `:first-child`, `:last-child` são exemplos. Elas permitem personalizar o estilo com base na posição ou tipo do elemento, contribuindo para um layout mais dinâmico e acessível.

Questão 17:

Layout responsivo adapta o conteúdo da página a diferentes tamanhos de tela. Media queries aplicam estilos condicionalmente com base em largura, altura, orientação ou resolução do dispositivo.

Questão 18:

Design fluido usa unidades relativas como `%`, `em`, `vw` e `vh`. Sistemas de grid facilitam a criação de layouts adaptáveis, organizando o conteúdo de forma consistente em qualquer resolução.