Lista 01 - 2º BIM - Desenvolvimento web I  
Sistemas de Informação | 1º Período - 2024  
Professor: Me. Jonathan Henrique Jeremias Souza

NOME: THALISON DE OLIVEIRA SANTOS R.A.: 003540 DATA: 03/06/2024

Orientações

**Entrega documento:** 03/06/2024 até as 23:59h - arquivo de texto (TXT) que contenha todos os códigos e respostas;  
**Valor:** 10 Pontos;  
**Tema:** DOM e seletores CSS, posicionamento, construção de layout.

1. Explique o que é DOM, e qual sua função em desenvolvimento web.

**R:** **DOM (Document Object Model)** é uma interface de programação que trata um documento HTML ou XML como uma árvore de objetos, onde cada nó da árvore representa uma parte do documento (elementos, atributos, texto, etc). No desenvolvimento web, o DOM permite que linguagens de programação, como JavaScript, acessem e manipulem a estrutura, estilo e conteúdo de uma página de forma dinâmica. Isso é fundamental para criar páginas web interativas e dinâmicas.

2. Explique o que são Seletores CSS e quais são suas funções em desenvolvimento web.

**R:** **Seletores CSS** são padrões usados para selecionar elementos HTML que queremos estilizar. Eles permitem aplicar estilos específicos a elementos individuais ou grupos de elementos em uma página web. A principal função dos seletores CSS é vincular regras de estilo aos elementos do DOM, permitindo que os desenvolvedores controlem a aparência visual das páginas de forma precisa e eficiente.

3. Explique e dê exemplos breves de aplicação dos principais tipos de seletores.

1. **Seletores de Tipo**: Selecionam todos os elementos de um tipo específico.

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

Seleciona todos os elementos <p>.

1. **Seletores de Classe**: Selecionam elementos com uma classe específica.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Seleciona todos os elementos com a classe "classe-exemplo".

1. **Seletores de ID**: Selecionam um elemento com um ID específico.

Texto

Descrição gerada automaticamente

Seleciona o elemento com o ID "id-exemplo".

1. **Seletores de Atributo**: Selecionam elementos com um atributo específico.

Texto

Descrição gerada automaticamente

Seleciona todos os <input> com type="text".

4. Explique e diferencie elementos de nível de bloco e inline.

R:

 **Elementos de Nível de Bloco**: Ocupam toda a largura disponível, começando em uma nova linha.

* Exemplo: <div>, <p>, <h1>

 **Elementos Inline**: Ocupam apenas o espaço necessário para o conteúdo e não iniciam uma nova linha.

* Exemplo: <span>, <a>, <strong>

5. Quais são os principais esquemas de posicionamento CSS?

Os **Principais Esquemas de Posicionamento CSS são:**

1. **Static;**
2. **Relative;**
3. **Absolute;**
4. **Fixed;**
5. **Sticky;**

6. Explique brevemente cada um dos esquemas de posicionamento citados na questão anterior.

R:

1. **Static**: Padrão, elementos são posicionados de acordo com o fluxo normal da página.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

1. **Relative**: Elementos são posicionados em relação à sua posição original.

Texto

Descrição gerada automaticamente

1. **Absolute**: Elementos são posicionados em relação ao seu elemento pai mais próximo com posicionamento diferente de static.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

1. **Fixed**: Elementos são posicionados em relação à janela de visualização (viewport).

Texto

Descrição gerada automaticamente

1. **Sticky**: Elementos alternam entre relative e fixed, dependendo da rolagem da página.

Texto

Descrição gerada automaticamente

7. Explique o que é viewport e como se aplica nos posicionamentos CSS.

**R:** **Viewport** é a área visível do usuário no navegador. Em CSS, propriedades como width, height, vw (viewport width) e vh (viewport height) são usadas para definir tamanhos e posicionamentos em relação ao viewport, garantindo que elementos se adaptem de acordo com a área visível do navegador.

8. Explique o que é posicionamento z-index, e qual sua aplicação.

**R:** **z-index** é uma propriedade CSS que controla a ordem de empilhamento dos elementos. Elementos com valores z-index maiores aparecem na frente daqueles com valores menores. É utilizado principalmente com posicionamentos relative, absolute, fixed e sticky.

Texto

Descrição gerada automaticamente

9. Explique o que é contexto de empilhamento.

**R: Contexto de Empilhamento** é o conceito que determina a ordem de sobreposição dos elementos na página. Cada elemento com um z-index e um posicionamento específico (diferente de static) cria um novo contexto de empilhamento, no qual seus filhos são empilhados em relação a ele.

10. Explique e diferencie layouts de largura fixa, líquidos, elásticos, híbridos e responsivos. Dê exemplos de aplicações para cada um destes, buscando apresentar suas vantagens e desvantagens.

R:

* **Largura Fixa**: Dimensões em pixels, não se adaptam a diferentes tamanhos de tela.
  + *Exemplo*: Sites antigos de desktop.
  + **Vantagens**: Consistência no design.
  + **Desvantagens**: Não são amigáveis para dispositivos móveis.
* **Líquidos**: Usam porcentagens para dimensões, adaptam-se ao tamanho da janela.
  + *Exemplo*: Sites básicos ajustáveis.
  + **Vantagens**: Flexibilidade.
  + **Desvantagens**: Pode haver problemas de legibilidade em tamanhos extremos.
* **Elásticos**: Usam unidades em para dimensões, adaptam-se ao tamanho da fonte.
  + *Exemplo*: Sites que dependem fortemente da tipografia.
  + **Vantagens**: Boa escalabilidade com o texto.
  + **Desvantagens**: Pode ser difícil de controlar o layout preciso.
* **Híbridos**: Combinam técnicas fixas e líquidas.
  + *Exemplo*: Layouts com colunas fixas e conteúdo fluido.
  + **Vantagens**: Flexibilidade combinada com controle.
  + **Desvantagens**: Complexidade no desenvolvimento.
* **Responsivos**: Usam media queries para adaptar o layout a diferentes tamanhos de tela.
  + *Exemplo*: Sites modernos (e.g., Bootstrap).
  + **Vantagens**: Otimização para qualquer dispositivo.
  + **Desvantagens**: Pode ser mais complexo de implementar.

Cada tipo de layout tem suas vantagens e desvantagens, e a escolha depende dos requisitos do projeto e da experiência do usuário desejada.