Aluno: Maria Claudia

Disciplina: Desenvolvimento Web - 2023-2

Professor: Eduardo Mangeli

Lista de Exercícios Desenvolvimento Web

1) Cite duas vantagens da utilização de sistemas de versionamento de código.

R: A utilização de versionamento pode evitar pode evitar problemas mantendo um histórico de alterações para a evolução de um software e a simplificação de um trabalho em equipe.

2) Explique a diferença entre git e github?

R: Git é a ferramenta de controle de versão que permite rastrear alterações em seu código, enquanto o GitHub é uma plataforma que utiliza o Git para hospedar repositórios de código-fonte e facilitar a colaboração entre desenvolvedores.

3) Quais as três principais linguagens utilizadas na construção de documentos da Web e quais são suas principais aplicações/funcionalidades?

R: HTML – corpo do documento, marcação semântica e de layout; CSS – formatação da apresentação, layout e animações; JavaScript – comportamento dinâmico, geração de conteúdo, comunicação e manipulação dos elementos do documento em ”tempo de execução”.

4) Qual é a diferença entre os conceitos de tag, elemento e atributo, no contexto do HTML?

R: Uma tag é um elemento fundamental no HTML e é usada para marcar partes específicas do conteúdo em uma página da web.

Um elemento é formado por uma ou mais tags (de abertura e fechamento ou apenas de abertura) e o conteúdo entre elas.

Um atributo é uma parte de uma tag que fornece informações adicionais sobre como um elemento deve ser processado ou exibido.

para criar uma estrutura HTML eficaz, você utiliza tags para marcar o conteúdo, cria elementos combinando tags de abertura e fechamento (ou tags auto-fecháveis) com conteúdo entre elas e usa atributos para fornecer informações adicionais sobre como os elementos devem ser tratados ou exibidos em uma página da web.

5) Desenvolva um documento HTML mínimo que contenha um parágrafo com a frase “Estou por aqui”, tenha o título “minha página” e use a codificação de caracteres utf-8.

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Minha Página</title>

</head>

<body>

<p>Estou por aqui</p>

</body>

</html>

6) O que é uma “pilha de protocolos”, no contexto da internet?

R: Uma "pilha de protocolos" na internet é um conjunto de regras organizadas em camadas para ajudar dispositivos a se comunicarem na internet de maneira organizada e confiável. Cada camada tem uma tarefa específica, como a camada de aplicação que lida com aplicativos, e a camada de transporte que cuida da entrega de dados. Isso torna a internet eficiente e funcional.

Simplificado: Uma "pilha de protocolos" na internet é um conjunto de regras que ajuda os computadores a conversarem entre si de maneira organizada. Isso faz com que a internet funcione direito.

7) Quais são os principais protocolos da camada de transporte da internet e qual a principal diferença entre eles?

Os dois principais protocolos da camada de transporte da Internet são o TCP (Transmission Control Protocol) e o UDP (User Datagram Protocol). A principal diferença entre eles está na forma como tratam a comunicação e na confiabilidade da entrega de dados:

* O TCP é um protocolo orientado à conexão e fornece uma comunicação confiável.
* Ele estabelece uma conexão entre os dispositivos de origem e destino antes de transmitir dados, garantindo que os dados sejam entregues na ordem correta e sem erros.
* O UDP é um protocolo sem conexão e não oferece garantia de entrega confiável.
* Ele é mais simples e rápido do que o TCP, pois não há overhead associado à confirmação de entrega e retransmissão de dados.

Simplificado:

* TCP é como enviar uma carta com confirmação de recebimento, para ter certeza de que chegará e na ordem certa.
* UDP é como enviar um cartão postal rápido, mas não tem confirmação, então pode se perder no caminho.

8) Quais as principais características da arquitetura cliente-servidor e como ela se diferencia de aplicações peer-to-peer?

R:A arquitetura cliente-servidor e as aplicações peer-to-peer são dois modelos de comunicação em rede amplamente utilizados, com características distintas. Vou explicar as principais características de cada um e destacar suas diferenças:

**Arquitetura Cliente-Servidor:**

* É como uma loja, onde os clientes fazem pedidos e os servidores fornecem os produtos.
* Os servidores são como gerentes, poderosos e centralizados, controlando os recursos.
* A comunicação acontece quando os clientes solicitam algo e os servidores respondem.
* Exemplos incluem a web, onde navegadores pedem páginas de servidores web, e e-mails, onde clientes de e-mail pegam mensagens de servidores de e-mail.

**Aplicações Peer-to-Peer (P2P):**

* É como uma rede de amigos, onde todos têm as mesmas responsabilidades e direitos.
* Não há chefes (servidores), todos são iguais, compartilhando recursos diretamente uns com os outros.
* A comunicação ocorre entre dispositivos diretamente, sem intermediários.
* Exemplos incluem compartilhamento de arquivos entre computadores, chamadas de vídeo entre telefones e redes blockchain, onde todos validam transações.

Diferença principal: Em cliente-servidor, há servidores poderosos que controlam tudo; em peer-to-peer, todos têm poder igual e trabalham juntos.

9) O que é um processo, no contexto do desenvolvimento de aplicações de internet, e como eles são identificados para efeitos de comunicação?

R:Um processo na internet é como um trabalho que um programa em um computador faz. Cada trabalho é identificado por um número de porta, como um endereço, para que outros programas saibam para onde enviar mensagens. Isso ajuda os programas em computadores diferentes a conversarem entre si. Por exemplo, quando você acessa um site, seu navegador (um programa) pede páginas a um servidor (outro programa) usando um número de porta específico e, assim, a informação é enviada e recebida corretamente.

10) Explique o que é SSL, no contexto da internet.

R:O SSL é um sistema de segurança para a internet. Ele torna difícil para outras pessoas verem informações que você envia online, como senhas ou números de cartão de crédito. Também ajuda a confirmar que você está se conectando ao site certo. Quando você vê um cadeado no navegador, significa que o SSL está em uso e suas informações estão protegidas. É importante para compras online, login seguro e outras atividades sensíveis na web.

11) Em um documento html, quantos são os filhos do nó raiz? Há limite para o número de descendentes do nó raiz? Justifique sua resposta

* R: Em um documento HTML, o nó raiz geralmente tem dois filhos diretos, que são as seções **<head>** e **<body>**. No entanto, não há limite para o número de elementos que podem estar dentro dessas seções. Você pode adicionar quantos elementos quiser, mas é importante manter o código organizado para facilitar a leitura e a manutenção.
* O número padrão de dois filhos diretos do nó raiz (elementos **<head>** e **<body>**) em um documento HTML é uma escolha convencional para organizar informações e conteúdo de forma clara. Isso ajuda a manter a página web organizada e fácil de entender. No entanto, você pode adicionar mais elementos conforme necessário para sua página, mas é importante manter a organização para que o código seja legível e fácil de manter.

12) Ao construir documentos html, é recomendado que usemos marcação que forneça significado (semântica) para os componentes do documento. Sendo assim, cite ao menos 5 tipos de elementos de marcação ou organização de conteúdo, além de 5 tipos de elementos de marcação semântica, explicando o significado de todos eles.

**R: Elementos de Organização de Conteúdo:**

1. **<div>**: É como uma caixa genérica para agrupar coisas.
2. **<span>**: É como uma mini-caixa para partes pequenas de texto.
3. **<ul>**: Cria uma lista desordenada.
4. **<ol>**: Cria uma lista ordenada.
5. **<table>**: É usado para fazer tabelas de informações.

**Elementos de Marcação Semântica:**

1. **<header>**: É o topo da página com coisas importantes.
2. **<nav>**: São os menus de navegação.
3. **<main>**: É o conteúdo principal da página.
4. **<article>**: É para partes independentes, como notícias ou posts de blog.
5. **<footer>**: É o rodapé com informações de contato ou direitos autorais.

Usar esses elementos ajuda a organizar e dar sentido ao conteúdo da página web.

13) Qual elemento é utilizado para que um documento html capture informações dos usuários?

R: Para capturar informações dos usuários em um documento HTML, você usa o elemento **<form>**. Ele cria um espaço onde os usuários podem inserir informações, como nomes e e-mails, e depois enviar esses dados. É como um formulário em papel, mas na web.

14) Qual elemento é utilizado para permitir a navegação entre diferentes documentos html?

R: Para permitir a navegação entre diferentes documentos HTML, você utiliza hiperlinks (links). Você cria um link usando a tag **<a>** e o atributo **href**, que aponta para o documento HTML que você deseja vincular. Quando os usuários clicam no link, eles são direcionados para esse documento.

15) Enumere ao menos cinco elementos utilizados em formulários html explicando sua utilidade.

R:

1. **<form>**: É o contêiner principal para todo o formulário.
2. **<input>**: Cria campos de entrada, como caixas de texto e botões.
3. **<label>**: Associa rótulos descritivos aos campos de entrada.
4. **<textarea>**: Permite inserir texto longo.
5. **<button>**: Cria botões para enviar ou realizar ações no formulário.

16) Explique o que são e para que são utilizados os seguintes atributos html:

a) class : É usada para aplicar estilos a um grupo de elementos semelhantes.

b) id: É usado para identificar exclusivamente um elemento, frequentemente usado em JavaScript ou CSS.

c) name: Usado em campos de formulário para nomear dados que serão enviados quando o formulário for enviado.

d) hidden: Oculta um elemento da página, mas ele ainda existe no código.

e) style: Aplica estilos diretamente a um elemento HTML.

f) title: Mostra uma dica de ferramenta quando você passa o mouse sobre o elemento.

17) O que seria “degradação graciosa”, também conhecida com “degradação elegante”?

R: "Degradação graciosa" significa criar websites ou aplicativos da web para que funcionem bem em diferentes navegadores e dispositivos, mesmo que alguns deles não suportem todos os recursos. Isso garante que todos os usuários tenham uma boa experiência, independentemente do que estão usando. É como ter um plano B para garantir que as coisas ainda funcionem quando algo não está disponível.

18) Diferencie e caracterize os conceitos de regra, seletor e declaração no contexto do CSS.

1. **Regra CSS**: É um conjunto de instruções que diz como estilizar elementos HTML. Ela contém um seletor (o que estilizar) e um bloco de declarações (como estilizar).
2. **Seletor CSS**: É a parte da regra que indica quais elementos HTML serão estilizados. Por exemplo, "p" estiliza todos os parágrafos.
3. **Declaração CSS**: É uma parte da regra que define uma propriedade de estilo (como cor de fundo) e o valor dessa propriedade (como vermelho).

Resumindo, uma regra CSS diz o que e como estilizar, com um seletor apontando para elementos e declarações definindo os estilos.

19) Diferencie os conceitos de “seletor de classe” e “seletor de id”, no contexto do CSS, fornecendo exemplos de cada um deles.

**Seletor de Classe (Class Selector)**:

* Usado para aplicar estilos a vários elementos com a mesma classe.
* Sintaxe: **.nomedaclasse**
* Exemplo: **.destaque** aplica estilo a todos os elementos com a classe "destaque" no HTML.

**Seletor de ID (ID Selector)**:

* Usado para aplicar estilos a um único elemento com um ID exclusivo.
* Sintaxe: **#nomedoid**
* Exemplo: **#cabecalho** aplica estilo a um elemento com o ID "cabecalho" no HTML.

Resumindo, classe é para vários elementos, ID é para um elemento único.

20) No contexto do CSS Box Model, explique os conceitos margin, border e padding

1. **Margin (Margem)**: É o espaço ao redor do elemento, criando afastamento entre ele e os elementos vizinhos.
2. **Border (Borda)**: É a linha visível ao redor do conteúdo do elemento, que separa o conteúdo da margem.
3. **Padding (Preenchimento)**: É o espaço entre o conteúdo do elemento e a borda, criando um espaço interno.

21) Explique e dê exemplos funcionais de utilização das propriedades de posicionamento: static, relative, absolute e fixed.

1. **Static (Estático)**: Comportamento padrão. Elementos aparecem na ordem do código, sem deslocamento.
2. **Relative (Relativo)**: Move o elemento em relação à sua posição normal, ocupando espaço original.
3. **Absolute (Absoluto)**: Remove o elemento do fluxo normal, posicionado em relação ao elemento pai mais próximo com posição não estática.
4. **Fixed (Fixo)**: Posiciona o elemento em relação à janela do navegador, mantendo-o fixo, mesmo durante a rolagem da página.

22) Como são definidos os blocos de comando no JavaScript?

No JavaScript, um bloco de comando é um conjunto de instruções agrupadas entre chaves **{}**. Esses blocos são usados para controlar a sequência de execução de código em estruturas como condicionais (**if**, **else**), loops (**for**, **while**) e funções.

Por exemplo, em uma estrutura condicional **if**:

if (condicao) {

// Este é um bloco de comando

// As instruções aqui são executadas se a condição for verdadeira

instrucao1;

instrucao2;

}

Em um loop **for**:

for (let i = 0; i < 5; i++) {

// Este é um bloco de comando

// As instruções aqui são repetidas enquanto a condição for verdadeira

instrucao1;

instrucao2;

}

Em uma função:

function minhaFuncao() {

// Este é um bloco de comando

// As instruções aqui são executadas quando a função é chamada

instrucao1;

instrucao2;

}

Os blocos de comando ajudam a organizar o código e definir o escopo das variáveis, permitindo um controle preciso sobre a execução das instruções no JavaScript.

23) Quais as diferenças entre as declarações de variáveis em JavaScript usando let, var e const?

* **let**: Variável com escopo de bloco (só visível dentro das chaves), mutável (pode ser atualizada).
* **var**: Variável com escopo de função (visível em toda a função), mutável.
* **const**: Variável com escopo de bloco, imutável (não pode ser reatribuída, mas valores internos de objetos ou arrays podem ser modificados).

Use **const** sempre que possível para tornar seu código mais seguro e previsível.