1. HTML, CSS e JavaScript
2. Explorando Sistemas Operacionais, Redes e Programação

João Marcos – DEV.B

Temas:

Tópicos Parte 1: Introdução ao HTML, Header e Body, Metatags, CSS, JavaScript e Estrutura de Decisão.

Tópicos Parte 2: Introdução, Switch/Case e Estruturas de Repetição, Sistemas Operacionais, CMD no Windows, Redes, Equipamentos e Meios de Conexão.

Introdução ao HTML, Header e Body e Metatags:

Tags Semânticas:

As tags semânticas são elementos no HTML introduzidos na versão HTML5 que fornecem significado e estrutura ao conteúdo da página, que também ajudam a descrever o tipo de informação que está contida em cada seção da página, facilitando a compreensão tanto para os desenvolvedores quanto para os mecanismos de busca. Principais tags semânticas:

1. “<header>”: Essa tag é usada para representar o cabeçalho da página ou de uma seção específica. Geralmente contém elementos como o logotipo, o título da página, menus de navegação e outros elementos relacionados ao topo da página.

2. “<nav>”: Essa tag é usada para definir uma seção de navegação. Eu e a maioria usam ela para agrupar links de navegação, menus, como “Início”, “Sobre”, etc.

3. “<main>”: Essa tag é usada para juntar o conteúdo principal da página. Contém o conteúdo principal bem “nítido” para o usuário.

4. “<section>”: Essa tag é usada para agrupar conteúdos qje tem relação dentro de uma página. Ela ajuda a organizar e estruturar o conteúdo, como uma gaveta no guarda-roupas.

7. “<footer>”: Essa tag é usada para representar o rodapé da página. Geralmente contém informações como direitos autorais, etc.

Estrutura básica do HTML:

Aqui está um exemplo de uma estrutura básica de um documento HTML:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Título</title>

</head>

<body>

<header>

melancia

</header>

<nav>

batata

</nav>

<main>

Joao

</main>

<footer>

Todos direitos reservados João M

</footer>

</body>

</html>

Explicação do “<!DOCTYPE html5>”:

“<!DOCTYPE html>” é a primeira linha no código HTML e indica a versão do HTML que estamos utilizando. No caso do exemplo acima, “<!DOCTYPE html5>” indica que o documento está sendo escrito em HTML5, isso é importante para garantir que o navegador interprete o documento da melhor forma.

Explicação da aplicação de ”<header>” e “<body>”:

A importância do uso da tag <header> está na criação de uma estrutura semântica para identificar claramente a seção de cabeçalho da página. O uso de tags semânticas ajuda a fornecer um significado mais claro e compreensível para o conteúdo da página, tanto para os desenvolvedores quanto para os mecanismos de busca.

O “<body>” é o elemento principal que contém todo o conteúdo visível da página. Ele junta todos os elementos que serão renderizados no navegador, como texto, imagens, links, vídeos etc. Todo o conteúdo visível para o usuário deve estar dentro da tag “<body>”.

Metatags:

As metatags são elementos do HTML colocados dentro da seção “<head>” de um documento HTML e fornecem informações adicionais sobre a página. Elas não são vistas por usuários, mas são lidas pelos mecanismos de busca e outros programas que analisam o conteúdo da página.

Elas são usadas para fornecer informações importantes sobre a página, como descrição, palavras-chave, autor, idioma, etc.

Um exemplo de metatag comum é: “<meta name="viewport" content="width=device-width, initial-scale=1.0">”, que é usada para controlar a forma como o navegador exibe e dimensiona o conteúdo em dispositivos móveis. Essa meta tag é especialmente importante para garantir uma experiência de visualização adequada em dispositivos com diferentes tamanhos de tela. Existem várias outras metatags disponíveis, cada uma com sua finalidade específica, e podem ser personalizadas de acordo com as necessidades de cada página.

W3C: (World Wide Web Consortium) é um consórcio internacional que tem como objetivo desenvolver padrões e diretrizes para a World Wide Web. Ele é composto por várias organizações, empresas e especialistas da área, que trabalham juntos para promover a interoperabilidade (capacidade de diferentes sistemas, dispositivos ou componentes interagirem e funcionarem em conjunto de maneira eficiente e sem problemas), e a evolução da web. A W3C desempenha um papel fundamental no desenvolvimento e na manutenção das especificações do HTML. Ela é responsável por estabelecer as normas e diretrizes para a linguagem, garantindo que ela seja consistente, acessível e compatível com diferentes navegadores e dispositivos.

O HTML (HyperText Markup Language) é uma linguagem de marcação utilizada para estruturar e apresentar o conteúdo em páginas da web. Basicamente é a base para a construção de sites e aplicativos web. A W3C desempenha um papel crucial na definição das especificações do HTML, garantindo que ele seja padronizado e interoperável.

Tags HTML mais utilizadas:

<html>: Define o início e o fim do documento HTML.

<head>: Contém informações sobre o documento, como metadados e links para folhas de *estilo.*

<body>: Contém todo o conteúdo visível da página.

<h1>, <h2>, <h3>, etc.: São usadas para criar cabeçalhos e títulos.

<p>: Utilizada para parágrafos de texto.

<a>: Cria um link para outra página ou recurso.

<img>: Insere uma imagem na página.

<div>: Usado como um container genérico para agrupar e estilizar elementos.

<ul>, <ol>, <li>: Utilizados para criar listas não ordenadas e ordenadas.

CSS (Cascading Style Sheets) é uma linguagem de estilo utilizada para definir a aparência e o layout de um documento HTML. Ela permite controlar o design visual dos elementos HTML, como cores, fontes, tamanhos, posicionamento e outros aspectos estilísticos.

As regras/propriedades mais utilizadas no CSS incluem:

color: Define a cor do texto.

font-family: Especifica a família de fontes a ser usada.

font-size: Define o tamanho da fonte.

background-color: Define a cor de fundo de um elemento.

margin: Controla as margens externas de um elemento.

padding: Controla as margens internas de um elemento.

width e height: Definem a largura e a altura de um elemento.

border: Especifica as propriedades da borda de um elemento.

display: Controla o tipo de exibição do elemento.

position: Define o posicionamento do elemento.

CSS significa "Cascading Style Sheets" e é uma linguagem de estilo usada para definir a aparência e o layout de um documento HTML. Ela separa o conteúdo da apresentação, permitindo um maior controle visual sobre os elementos HTML.

Tipos de seletores CSS:

Seletor de elemento: Seleciona elementos com base em seus nomes de tag.

Exemplo: h1, p, div.

Seletor de classe: Seleciona elementos com base em suas classes atribuídas. Exemplo: .classe.

Seletor de ID: Seleciona um elemento com base em seu ID exclusivo. Exemplo: #id.

Seletor descendente: Seleciona elementos filhos de um elemento pai específico. Exemplo: pai filho.

Seletor de pseudo-classe: Seleciona elementos com base em um estado ou interação específica. Exemplo: :hover, :nth-child.

Especificidade: HTML: <div class="conteudo"> <p class="texto">Texto de exemplo</p> </div>

CSS: .conteudo p { color: red; } .texto { color: blue; }

Nesse exemplo, a cor do texto será vermelha porque o seletor de classe combinado com o seletor de elemento “(.conteudo p)” possui uma especificidade maior do que o seletor de classe “(.texto)”.

CSS pode ser utilizado de três formas:

CSS Inline: O código CSS é inserido diretamente nos atributos HTML. Exemplo: “<p style="color: blue;">Texto de exemplo</p>”. Não gosto de trabalhar com esse estilo, muito confuso e desorganizado.

CSS incorporado: O código CSS é inserido dentro das tags <style> no cabeçalho do documento HTML. Exemplo:

“<head> <style> p { color: blue; } </style> </head>”, que também não acho prático trabalhar.

CSS externo: O código CSS é armazenado em um arquivo separado, vinculado ao documento HTML usando a tag <link>. Exemplo:

<head> <link rel="stylesheet" href="exemplo.css"> </head>

Em minha opinião, a mais prática e organizada.

JavaScript é uma linguagem de programação de alto nível, interpretada e orientada a objetos. Ela é amplamente utilizada para desenvolvimento web, permitindo a criação de páginas dinâmicas e interativas.

Linguagem interpretada significa que o código JavaScript é executado linha por linha por um interpretador (como o navegador), sem a necessidade de compilação prévia. Isso permite uma execução mais flexível e rápida do código, mas também pode resultar em uma menor velocidade de execução em comparação com linguagens compiladas.

Para declarar variáveis em JavaScript, usamos a palavra-chave "var", "let" ou "const", seguida pelo nome da variável e, opcionalmente, atribuímos um valor a ela usando o operador de atribuição "=".

Exemplo de declaração de variáveis:

var nome = "João"; let idade = 25; const PI = 3.14;

JavaScript possui diferentes tipos de dados, incluindo:

Números (Numbers): Representam valores numéricos. Podem ser inteiros ou decimais.

let idade = 25; let peso = 68.5;

Strings: Representam sequências de caracteres, delimitadas por aspas simples ou duplas.

let nome = "Maria"; let mensagem = 'Olá, mundo!';

Booleanos (Boolean): Representam valores verdadeiro (true) ou falso (false).

let estudante = true; let empregado = false;

Arrays: São coleções ordenadas de valores, separados por vírgulas e delimitados por colchetes.

let frutas = ["maçã", "banana", "laranja"];

Objetos (Objects): São estruturas de dados que armazenam propriedades e seus valores relacionados.

let pessoa = { nome: "João", idade: 30, profissao: "engenheiro" };

Para usar a convenção de nomes em JavaScript, geralmente segue-se as seguintes práticas:

Operadores Aritméticos:

Soma (+): Realiza a adição de dois valores.

let resultado = 5 + 3; // resultado terá o valor 8

Subtração (-): Realiza a subtração de dois valores.

let resultado = 10 - 5; // resultado terá o valor 5

Multiplicação (\*): Realiza a multiplicação de dois valores.

let resultado = 4 \* 3; // resultado terá o valor 12

Divisão (/): Realiza a divisão de dois valores.

let resultado = 20 / 5; // resultado terá o valor 4

PRIORIDADE: A prioridade de operador aritmético define a ordem em que as operações matemáticas são executadas em uma expressão. A regra geral é que as operações de multiplicação, divisão e módulo têm prioridade sobre as operações de adição e subtração. Parênteses podem ser usados para alterar a prioridade de execução.

Operadores Lógicos:

AND (&&): Retorna verdadeiro se ambas as expressões forem verdadeiras.

let idade = 25; let estudante = true; if (idade >= 18 && estudante) { console.log("É maior de idade e estudante."); }

OR (||): Retorna verdadeiro se pelo menos uma das expressões for verdadeira.

let idade = 16; let estudante = false; if (idade >= 18 || estudante) { console.log("É maior de idade ou estudante."); }

NOT (!): Inverte o valor lógico de uma expressão.let estudante = false; if (!estudante) { console.log("Não é estudante."); }

Estrutura de Decisão: Exemplo de uso do IF/ELSE/ELSE IF:

let idade = 20; if (idade < 18) { console.log("Menor de idade."); } else if (idade >= 18 && idade < 65) { console.log("Adulto."); } else { console.log("Idoso."); }

Exemplo de uso do Switch/Case:

let diaDaSemana = 2; let mensagem; switch (diaDaSemana) { case 1: mensagem = "Hoje é segunda-feira."; break; case 2: mensagem = "Hoje é terça-feira."; break; case 3: mensagem = "Hoje é quarta-feira."; break; default: mensagem = "Dia da semana desconhecido."; } console.log(mensagem);

Estrutura de Repetição: Exemplo do uso do While:

let contador = 0; while (contador < 5) { console.log(contador); contador++; }

Exemplo do uso do Do-While:

let contador = 0; do { console.log(contador); contador++; } while (contador < 5);

Exemplo do uso do For:

for (let i = 0; i < 5; i++) { console.log(i); }

SISTEMA OPERACIONAL:

Um sistema operacional é um software responsável por gerenciar os recursos de hardware e software de um computador e fornecer uma interface entre o usuário e o computador. Ele controla o acesso aos dispositivos, permite a execução de programas e fornece serviços que são essenciais para o funcionamento do sistema.

Existem diferentes sistemas operacionais disponíveis, cada um com suas próprias características e funcionalidades. Alguns exemplos incluem:

- Windows: Desenvolvido pela Microsoft, o Windows é um sistema operacional amplamente utilizado em computadores pessoais. Ele fornece uma interface gráfica do usuário (GUI) e suporta uma variedade de aplicativos e jogos.

- macOS: Desenvolvido pela Apple, o macOS é o sistema operacional usado em computadores Mac. Ele oferece uma experiência integrada com outros dispositivos da Apple e é conhecido por sua interface intuitiva e recursos avançados de produtividade.

- Linux: O Linux é um sistema operacional de código aberto baseado no kernel do Unix. Existem várias distribuições de Linux disponíveis, como Debian, Ubuntu, Fedora, entre outras. O Linux é conhecido por sua estabilidade, segurança e flexibilidade, sendo amplamente utilizado em servidores e dispositivos embarcados.

- Android: Desenvolvido pelo Google, o Android é um sistema operacional móvel baseado em Linux. Ele é usado em uma ampla gama de dispositivos, como smartphones, tablets e TVs inteligentes.

Código Fechado (Proprietário) refere-se a software cujo código-fonte não é disponibilizado publicamente. A empresa ou indivíduo que detém os direitos autorais do software é responsável por sua criação, distribuição e manutenção. Os usuários do software não têm acesso ao código-fonte e não podem modificá-lo livremente.

Código Aberto refere-se a software cujo código-fonte é disponibilizado publicamente e pode ser modificado e distribuído livremente por qualquer pessoa. O código aberto promove a colaboração e transparência, permitindo que desenvolvedores e usuários contribuam para o aprimoramento do software.

Como criar o repositório na VM (Virtualbox) com a distro Debian e no Windows CMD:

Demonstração no VM (Virtualbox) - Distro Debian:

1. Abra a VM e inicie a distro Debian.

2. Abra um terminal.

3. Digite o seguinte comando para criar o diretório "reposicao" em /home:

mkdir /home/reposicao

Demonstração no Windows CMD:

1. Abra o CMD no Windows.

2. Navegue até o diretório C:/Users utilizando o comando "cd":

cd C:\Users, ou, usar o comando “cd..” até chegar onde deseja.

3. Para listar os subdiretórios e arquivos presentes no diretório atual, utilize o comando "dir":

dir

Topologia de rede se refere à forma como os dispositivos de uma rede estão interconectados. Principais topologias:

1. Topologia em Estrela: Nessa topologia, todos os dispositivos são conectados a um único ponto central, geralmente um switch. Os dispositivos se comunicam através desse ponto central, permitindo uma fácil administração da rede.

2. Topologia em Barramento: Nessa topologia, todos os dispositivos são conectados a um cabo compartilhado, conhecido como barramento. Os dados são enviados através do barramento e cada dispositivo recebe os dados destinados a ele.

3. Topologia em Anel: Nessa topologia, os dispositivos são conectados em uma forma de anel fechado, onde cada dispositivo está conectado ao dispositivo adjacente. Os dados são transmitidos em um único sentido através do anel.

Equipamentos de rede:

- Roteador: Um roteador é um dispositivo responsável por encaminhar os pacotes de dados entre diferentes redes. Ele atua como um ponto de conexão entre a rede local e a internet, permitindo o compartilhamento de uma conexão de internet entre vários dispositivos.

- Modem: Um modem é um dispositivo utilizado para estabelecer a conexão com a internet. Ele converte os sinais digitais dos dispositivos em sinais analógicos que podem ser transmitidos pela linha telefônica, fibra ótica ou outro meio de comunicação, e vice-versa.

- Switch/Hub: Switches e hubs são dispositivos utilizados para interconectar vários dispositivos em uma rede local. Um switch é mais inteligente e eficiente, pois direciona os pacotes de dados apenas para o dispositivo de destino, enquanto um hub simplesmente repete os dados para todos os dispositivos conectados.

Meio de conexão:

- Fibra ótica: A fibra ótica é um meio de conexão que utiliza cabos de fibra ótica para transmitir sinais de luz. Ela oferece alta velocidade de transmissão, largura de banda ampla e é imune a interferências eletromagnéticas.

- Par trançado: O par trançado é um meio de conexão que utiliza pares de fios de cobre trançados para transmitir sinais elétricos. É amplamente utilizado em redes locais (LANs) e pode ser encontrado em duas formas principais: par trançado não blindado (UTP) e par trançado blindado (STP).

Para verificar o IP da máquina local no CMD, utilizo o comando "ipconfig".

1. Abra o CMD.

2. Digite o seguinte comando:

“Ipconfig”

IP da máquina 3-10: 192.168.56.1