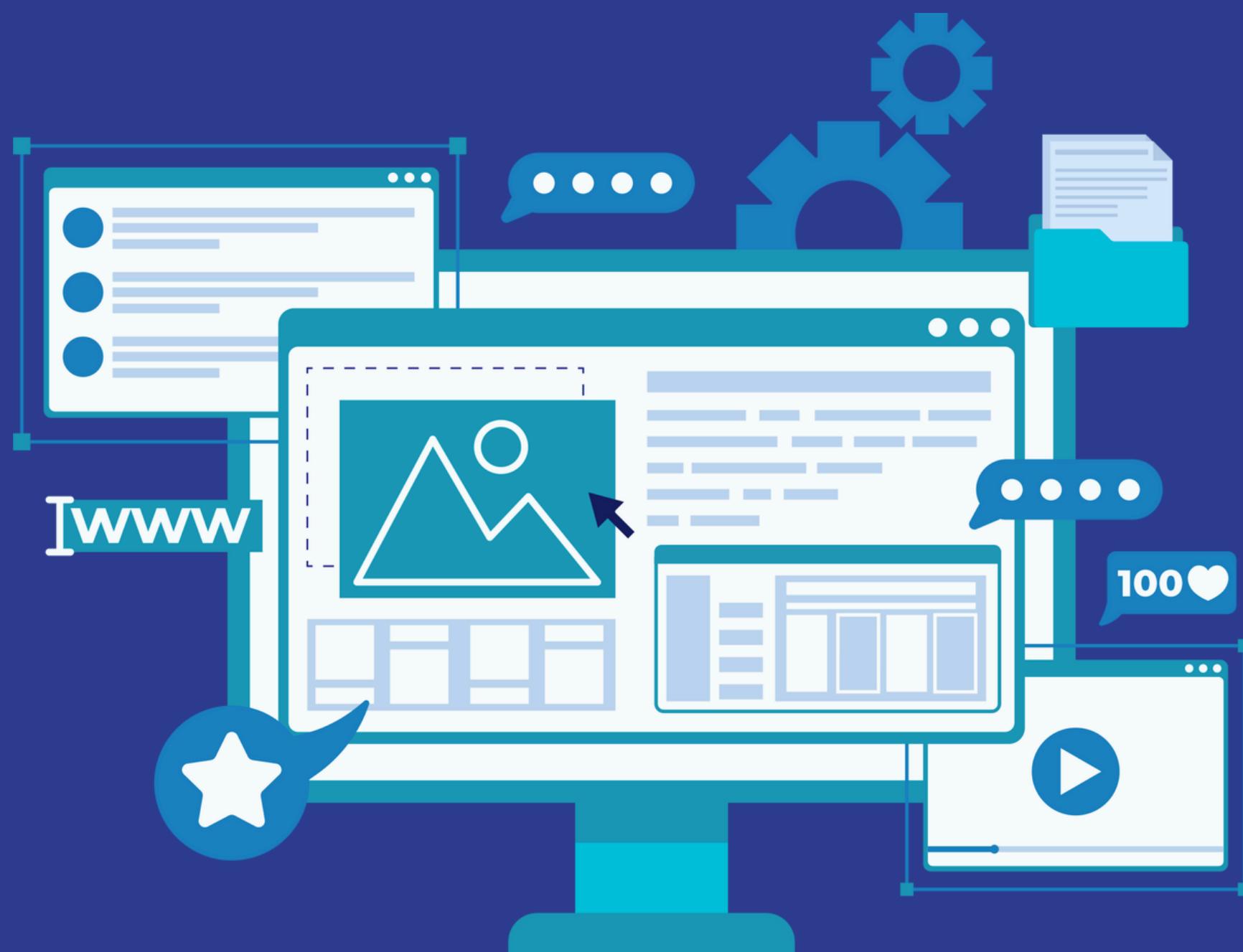


E-BOOK

Desenvolvimento Web

Primeiro Site: Principais Ferramentas para Iniciantes



DESENVOLVIDO POR:

Alice Matos
Davi José Carvalho
Fernanda Dias
João Gabriel Souza

REVISADO POR:

João Paulo Rabelo



SUMÁRIO

Introdução.....	4
Internet	5
HTML.....	10
CSS	15
JavaScript	20
Gabarito	25

INTRODUÇÃO

Neste e-book vamos explorar as principais linguagens de marcação, estilização e programação usadas no desenvolvimento web: HTML, CSS e Java Script.

Com explicações claras e exemplos práticos, você vai aprender os conceitos mais abordados no frontend.

Indicado principalmente para os iniciantes, este guia vai ajudá-lo a compreender as ferramentas essenciais para lançar seu próprio projeto na web e criar o seu primeiro site funcional.

Ao final, você terá uma base para personalizar suas criações, se preparar para projetos maiores e dar os primeiros passos no mundo do desenvolvimento web. Ideal para quem deseja transformar suas ideias em realidade digital!



MÓDULO 1

COMO A INTERNET FUNCIONA?

O QUE É A INTERNET? E QUAIS SÃO AS VANTAGENS DE USÁ-LA?

A Internet é uma interconexão global de computadores que utilizam o **Protocolo de Internet** (IP), conhecida como uma "rede de redes". Para entender essa definição, é necessário compreender os componentes de uma rede:

- **Dispositivos:** Equipamentos que se conectam e se comunicam na rede.
- **Transmissão:** Os meios pelos quais os dispositivos se conectam, como cabos físicos (Ethernet), fibras ópticas e conexões sem fio (Wi-Fi, Bluetooth).
- **Roteadores e Switches:** Equipamentos que gerenciam o tráfego de dados, com roteadores direcionando o tráfego entre redes diferentes e switches conectando dispositivos dentro da mesma rede.
- **Protocolos:** Conjuntos de regras para transmissão de dados, como o IP e o TCP (Protocolo de Controle de Transmissão).

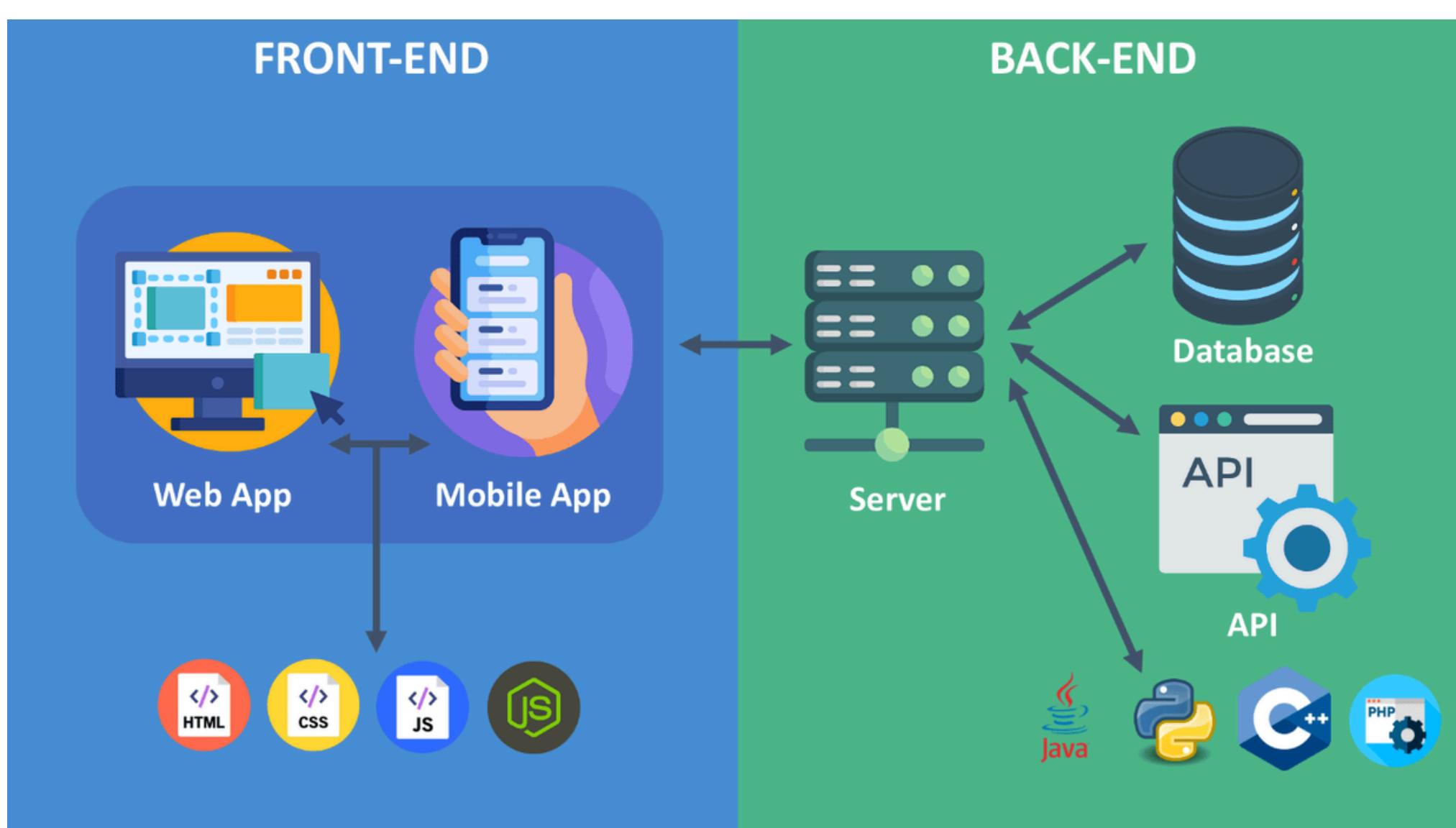
A Internet é uma rede **WAN** (Rede de Longa Distância), conectando várias redes locais e globais, permitindo o compartilhamento de informações e a comunicação em tempo real entre pessoas e organizações ao redor do mundo.

WEBSITES E DESENVOLVIMENTO WEB

Um website é um **conjunto de páginas** interligadas sob um único domínio, acessadas por navegador via URL. As páginas, criadas em HTML, usam CSS para estilos visuais e JavaScript para interatividade. Websites podem ter conteúdo **estático** (fixo) ou **dinâmico** (personalizado conforme interação do usuário ou banco de dados).

O desenvolvimento web divide-se em duas áreas principais:

- **Frontend:** Cuida da interface visual e interativa com a qual o usuário interage, usando HTML, CSS e JavaScript, além de frameworks e bibliotecas.
- **Backend:** Responsável por processar dados e gerenciar a comunicação entre o servidor e a aplicação web, garantindo seu funcionamento adequado.



COMPONENTES FUNDAMENTAIS DA WEB:

HTTPS, DNS, DOMÍNIOS, SERVIDORES E HOSPEDAGEM

Quando você digita o endereço de um site no navegador:

- **DNS** é acionado para encontrar o endereço IP do servidor associado ao nome de domínio.
- O navegador usa **HTTPS** para estabelecer uma conexão segura com o servidor onde o site está hospedado.
- O **servidor** responde à solicitação, enviando a página web solicitada, que o navegador renderiza.

DNS (Domain Name System)

O **DNS** converte nomes de **domínio** legíveis (como www.exemplo.com) em **endereços IP** (como 192.0.2.1), que são usados por computadores para localizar e se comunicar na Internet. Funciona como uma "agenda telefônica", permitindo que o navegador encontre o servidor correto para acessar um site.

Domínios

Um domínio é o **nome atribuído** a um endereço web, facilitando o acesso sem a necessidade de lembrar endereços IP. Ele é composto pelo **nome de domínio** (ex: exemplo) e o **TLD** (ex: .com, .org). Para funcionar, o domínio precisa ser registrado e associado a um DNS que aponte para o servidor do site.

COMPONENTES FUNDAMENTAIS DA WEB:

HTTPS, DNS, DOMÍNIOS, SERVIDORES E HOSPEDAGEM

HTTPS (HyperText Transfer Protocol Secure)

O HTTPS é um **protocolo seguro** para a transferência de dados entre o navegador e o servidor de um site. Ele utiliza **SSL/TLS** para criptografar a comunicação, protegendo informações pessoais e senhas contra interceptação. Quando um site usa HTTPS, o navegador exibe um cadeado, indicando uma conexão segura.

Servidores

Um servidor web **armazena e entrega** os arquivos e dados de um site aos usuários. Ele **processa** as requisições do navegador e **responde** com o conteúdo correto. Há diferentes tipos de servidores:

- **Físicos:** Máquinas localizadas em data centers.
- **Virtuais (VPS):** Divisões de servidores físicos que operam de forma independente.
- **Nuvem:** Recursos escaláveis em múltiplos servidores conectados em rede.



COMPONENTES FUNDAMENTAIS DA WEB:

HTTPS, DNS, DOMÍNIOS, SERVIDORES E HOSPEDAGEM

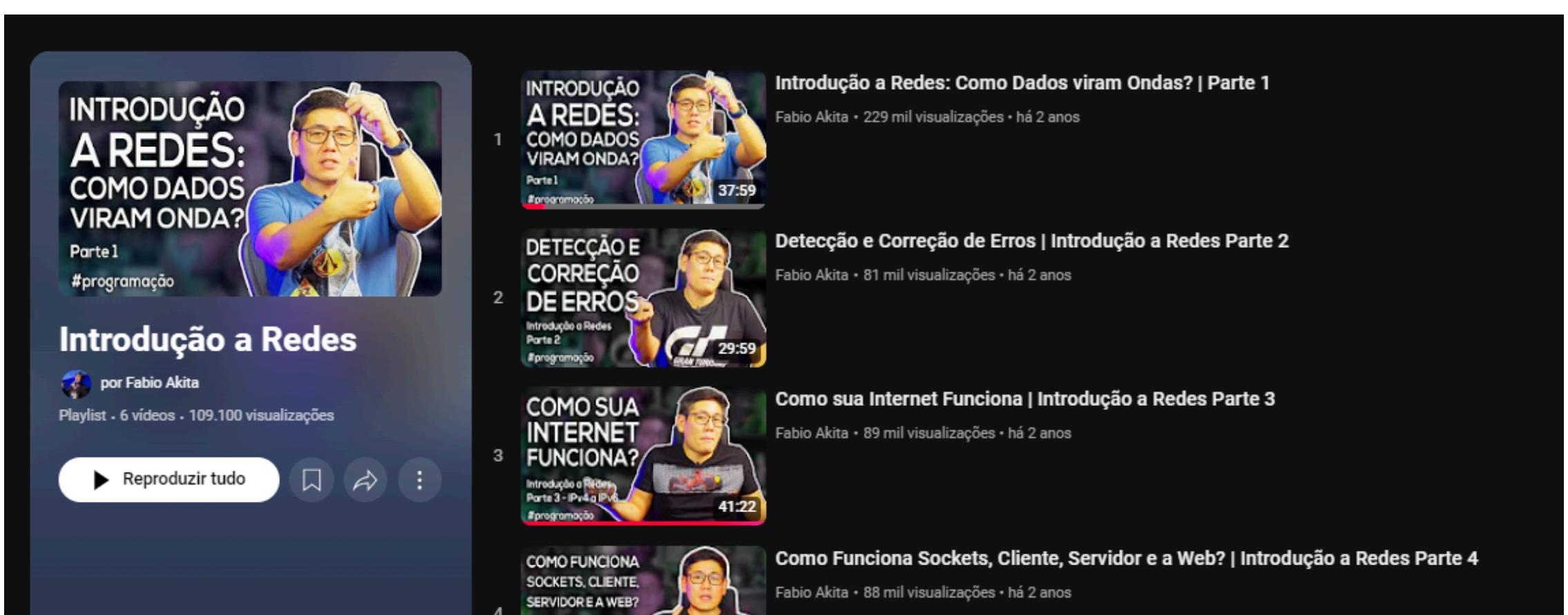
Hospedagem

A hospedagem web é o serviço que permite **armazenar um site em um servidor** e torná-lo acessível pela Internet. Tipos de hospedagem incluem:

- **Compartilhada:** Vários sites no mesmo servidor.
- **VPS:** Servidor virtual com mais controle e recursos.
- **Dedicada:** Um servidor inteiro reservado para um único site.
- **Nuvem:** Utiliza vários servidores, proporcionando flexibilidade e escalabilidade.

Se algum desses elementos falhar (por exemplo, um problema com o DNS ou um certificado HTTPS inválido), o site pode se tornar inacessível ou inseguro para os usuários.

Sugestão para aprimorar seus conhecimentos:



Link: <https://www.youtube.com/playlist?list=PLdsnXVqbHDUcTGjNZuRYCVj3AZtdt6oG7>

PRATIQUE E APLIQUE OS CONHECIMENTOS ADQUIRIDOS NESTE MÓDULO.

1 - Como uma página web é geralmente dividida em termos de estrutura e funcionamento?

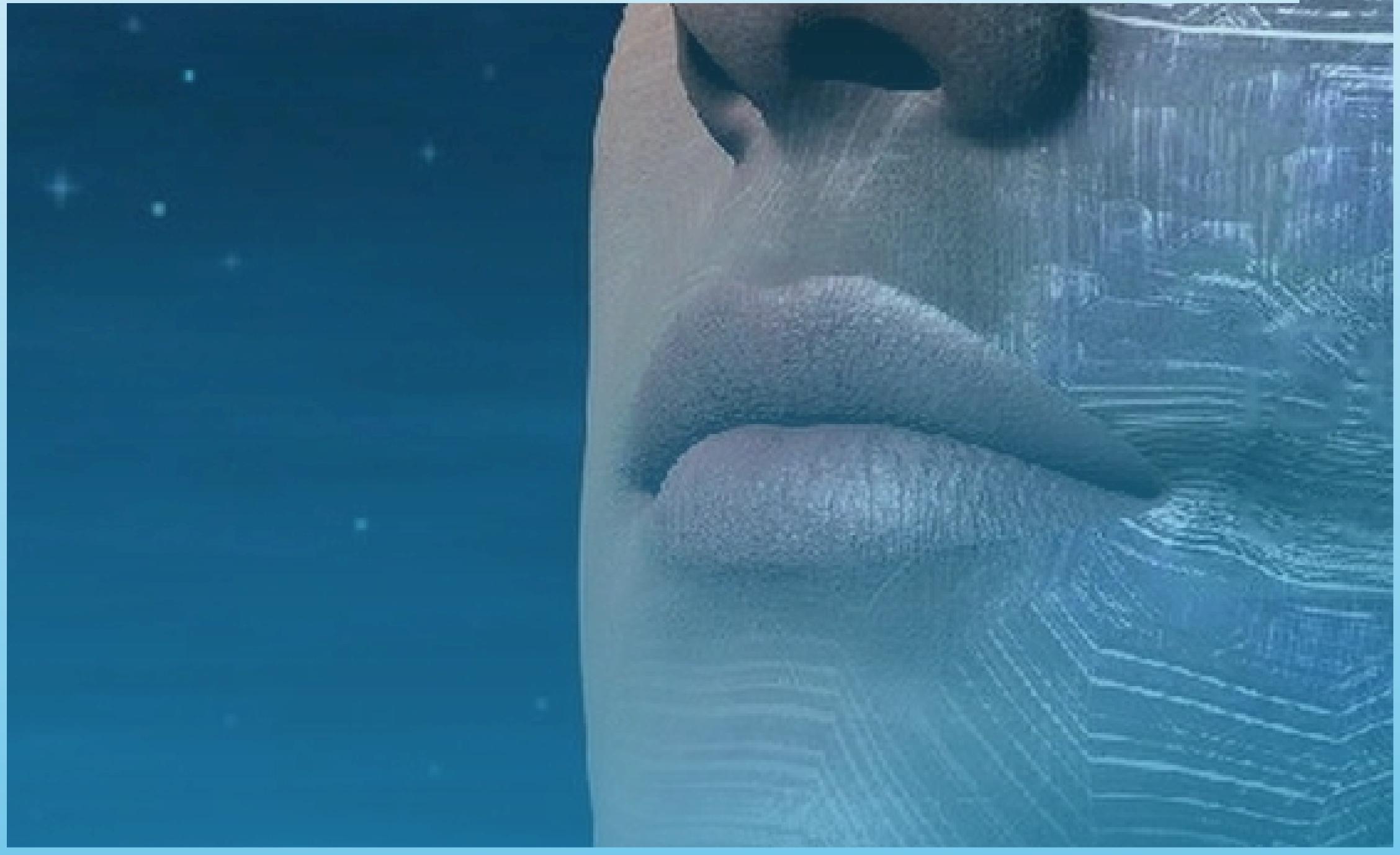
- a) Frontend e Backend
- b) HTML, CSS e JavaScript
- c) Domínio, hospedagem e arquivos
- d) Cliente e Servidor

2 - Qual das alternativas não descreve corretamente a função de um componente da rede na Internet?

- A) Dispositivos são equipamentos que enviam e recebem dados na rede.
- B) Roteadores conectam diferentes redes e gerenciam o tráfego entre elas.
- C) TCP garante a entrega confiável de pacotes de dados na rede.
- D) Switches conectam redes locais diferentes, permitindo a comunicação entre elas.

3 - Explique o papel dos roteadores e switches na comunicação de dados em redes, destacando as diferenças entre suas funções.

Nota: Anote suas respostas em um bloco de notas e consulte o gabarito na última página.



MÓDULO 2

FUNDAMENTOS DO HTML

INTRODUÇÃO AO HTML

HTML, ou Linguagem de Marcação de Hipertexto, é a **base para criar e estruturar** páginas na web. Diferente das linguagens de programação, HTML organiza o conteúdo de um site usando **tags e elementos**, que indicam ao navegador como exibir textos, imagens, vídeos e outros conteúdos. Cada página HTML é composta por uma série de tags, como <p>, que define parágrafos.

A Estrutura de uma Página Web

A estrutura de uma página web **organiza** o conteúdo exibido aos usuários e é construída principalmente com HTML, que fornece a base para criação de páginas. Ela inclui componentes essenciais que definem a apresentação e organização do conteúdo no navegador.

Abaixo, detalharemos os **componentes essenciais** que constituem essa estrutura:

```
DOCTYPE html>
html lang="pt-br">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Página Web</title>

  <link rel="stylesheet" href="style.css">
</head>
<body></body>
html>
```

ESTRUTURA DE UMA PÁGINA WEB

Todo documento HTML segue uma estrutura básica para que o navegador interprete corretamente o conteúdo. A estrutura típica inclui:

- **<!DOCTYPE html>**: Informa ao navegador que o arquivo usa HTML5.
- Tag **<html>**: Envolve todo o conteúdo da página, indicando que é um documento HTML.
- Seção **<head>**: Contém metadados, como o título da página, links para CSS e scripts JavaScript.
- Seção **<body>**: Contém o conteúdo visível da página, incluindo textos, imagens e outros elementos.

No Visual Studio Code, o **atelho "!"** gera automaticamente essa estrutura básica.

```
① index.html > ⚙️ html
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Trilha de Aprendizado</title>
7     </head>
8     <body>
9       <h1>Introdução ao Html</h1>
10      <h2>Todo sobre como começar um site</h2>
11    </body>
12  </html>
```

ELEMENTOS DE PARÁGRAFOS E TÍTULOS

Os elementos de parágrafo e título são essenciais para organizar o conteúdo de uma página web, melhorando a legibilidade e a semântica. Títulos de **<h1>** a **<h6>** definem a hierarquia do conteúdo:

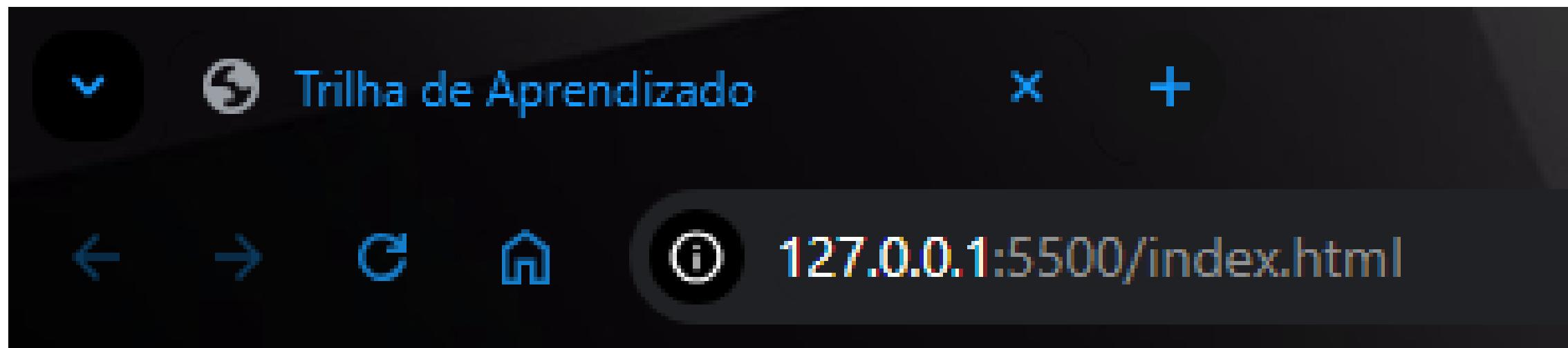
- **<h1>**: título principal da página, usado uma única vez.
- **<h2>**: subtítulo de seções principais.
- **<h3>** a **<h6>**: subdivisões menores, com tamanho de texto diminuindo conforme o número da tag aumenta.

Esses elementos estruturam o texto, destacando temas centrais e subdivisões.

```
8  <body>
9    <h1>Introdução ao Html</h1>
10   <h2>Todo sobre como começar um site</h2>
11   <h3>Olí Mundo</h3>
12   <h4>Ebook</h4>
13   <h5>Treinamento</h5>
14
15 </body>
```

ELEMENTOS DE PARÁGRAFOS E TÍTULOS

Utilizando o código anterior, temos esta página:



Introdução ao Html

Tudo sobre como começar um site

Oi Mundo

Ebook

Treinamento

Apesar de ser uma página simples, podemos utilizar mais tags para adicionar mais informações ao site.

ELEMENTOS DE FORMATAÇÃO

Os elementos de parágrafo são representados pela tag (**<P>**). Eles são usados para agrupar blocos de texto, tornando-o mais legível e organizado. Cada parágrafo deve conter uma ideia ou tema principal, facilitando a compreensão do conteúdo.

```
<p>Vamos para a etapa 2 desse projeto</p>
```

Elementos de Formatação

Elementos Inline

Os elementos inline são usados para formatar partes do conteúdo sem quebrar o fluxo do texto. Eles geralmente não iniciam uma nova linha e podem ser aninhados dentro de elementos de bloco. Alguns dos principais elementos inline incluem:

1. ****: indica que o texto deve ser fortemente enfatizado, geralmente renderizado em negrito. Também pode ser escrito utilizando a tag '****'.

```
<p>Esse é um texto <strong>Importante</strong>.</p>
<p>Esse é um texto <b>negrito</b>.</p>
```

ELEMENTOS DE FORMATAÇÃO

2. ****: indica ênfase no texto, geralmente renderizado em itálico. Também pode ser escrito utilizando a tag '**<i>**'.

```
<p>Esse é um texto <em>enfatizado</em>. </p>
<p>Esse é um texto <i>itálico</i>. </p>
```

3. **<a>**: cria um hiperlink, permitindo que os usuários naveguem para outra página ou recurso.

```
<p>Visite meu site em <a href="https://example.com"> exemplo.com </a></p>
```

Elementos de Quebra de Linha

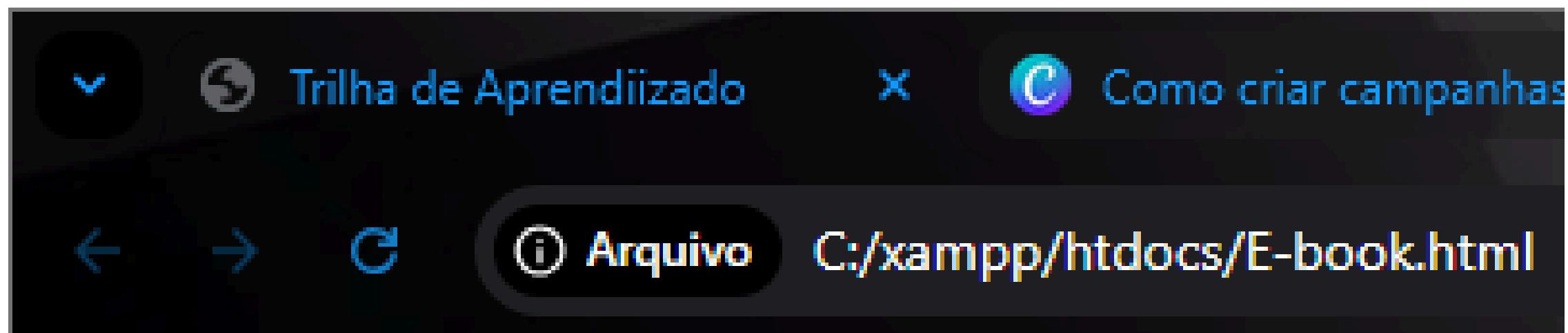
Também existem os elementos de quebra de linha, que fazem o texto ser dividido na página.

1. **
**: Insere uma quebra de linha no texto. É um elemento vazio e não requer tag de fechamento.
2. **<hr>**: insere uma linha horizontal, frequentemente usada para separar seções de conteúdo.

```
<p>Está é uma linha. <br />Está é outra linha.</p>
<hr />
```

ELEMENTOS DE FORMATAÇÃO

Esta é a aparência da nossa página com os elementos de formatação:



Introdução ao Html

Tudo sobre como começar um site

Oi Mundo

Ebook

Treinamento

Esse é um texto importante.

Esse é um texto negrito.

Esse é um texto enfatizado.

Esse é um texto itálico.

Visite meu site em exemplo.com

Está é uma linha.

Está é outra linha.

ELEMENTOS SEMÂNTICOS

Além da estrutura básica, a utilização de elementos semânticos é essencial para **melhorar a acessibilidade e SEO** (otimização para motores de busca). Esses elementos ajudam a definir partes específicas do conteúdo, como:

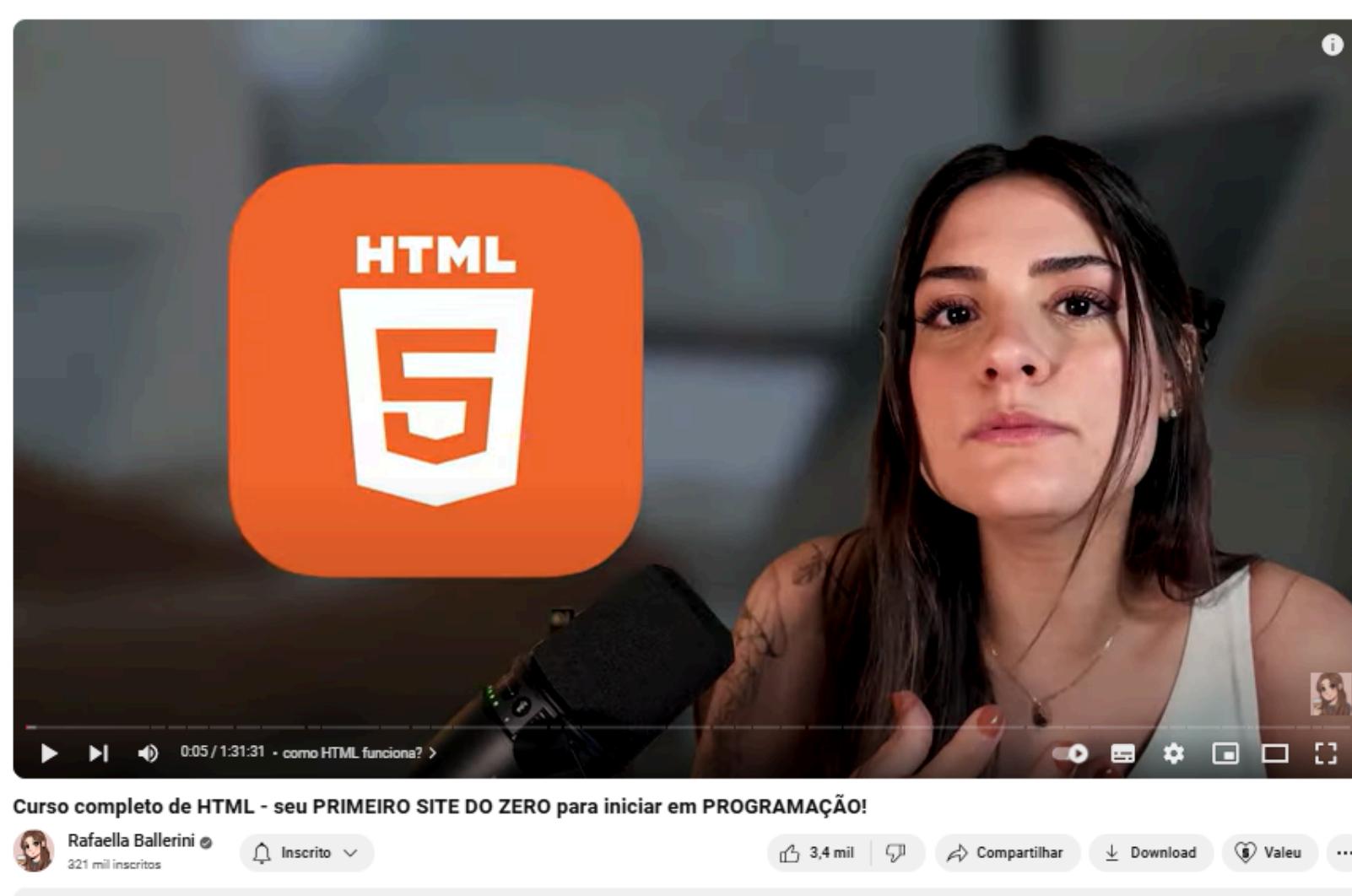
<header>: representa o cabeçalho da página ou seção.

<nav>: define uma seção de navegação.

<main>: indica o conteúdo principal da página.

<footer>: representa o rodapé da página ou seção.

Sugestão para aprimorar seus conhecimentos em HTML:



Link: [https://www.youtube.com/playlist?
list=PLdsnXVqbHDUcTGjNZuRYCVj3AZtdt6oG7](https://www.youtube.com/playlist?list=PLdsnXVqbHDUcTGjNZuRYCVj3AZtdt6oG7)

PRATIQUE E APLIQUE OS CONHECIMENTOS ADQUIRIDOS NESTE MÓDULO.

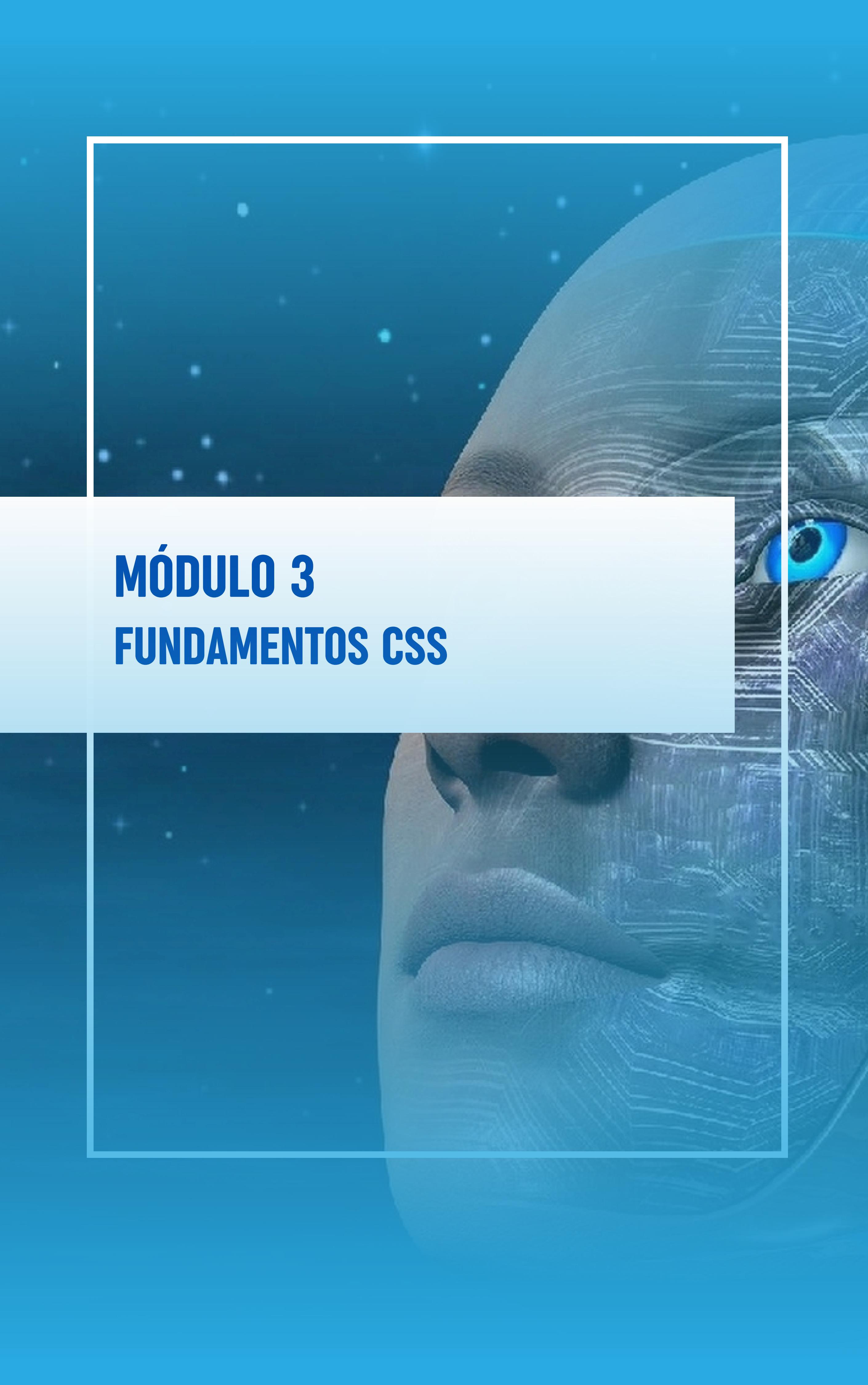
1 - Qual das seguintes tags HTML é usada para definir o título principal de uma página web?

- a) <title>
- b) <h1>
- c) <header>
- d) <p>

2 - Qual é a função da tag em HTML?

- a) Exibir o conteúdo principal da página
- b) Definir a área de rodapé
- c) Criar uma seção de navegação
- d) Inserir uma barra lateral

3 - Explique a importância de uma estrutura semântica no HTML e cite dois elementos semânticos que ajudam na acessibilidade e SEO de uma página.



MÓDULO 3

FUNDAMENTOS CSS

OQUE É CSS?

CSS, sigla para Cascading Style Sheets ou Folhas de Estilo em Cascata, é uma linguagem de estilo utilizada para **formatar a apresentação** de documentos escritos em HTML. Em outras palavras, o CSS define **como os elementos HTML serão exibidos na tela**, separando a estrutura da aparência do site. Tendo o objetivo de estilizar a pagina.

Por que usar CSS?

Separação de conteúdo e apresentação: Facilita a manutenção do código, tornando-o mais organizado e legível.

Reutilização de estilos: Permite aplicar o mesmo estilo a vários elementos, economizando tempo e esforço.

Melhora na acessibilidade: ao aplicar estilos de forma consistente, você garante uma experiência mais acessível para usuários com diferentes necessidades.

Responsividade: O CSS possibilita a criação de layouts adaptáveis a diferentes tamanhos de tela, otimizando a visualização em dispositivos móveis e desktops.

ESTRUTURA BÁSICA DE UM ARQUIVO CSS

Um arquivo CSS é composto **por seletores e declarações**.

Seletores: Identificam os elementos HTML que serão estilizados.

Declarações: Definem as propriedades e valores que serão aplicados aos elementos selecionados.

<!-- end list -->

```
/* Selecionando todos os elementos de parágrafo */
p {
    color: blue;
    font-size: 16px;
}
```

SELETORES CSS

Existem diversos tipos de seletores, cada um com sua função específica:

Tipo: Selecionam elementos pelo seu nome de tag (**e.g., p, h1, div**).

Classe: Selecionam elementos com uma determinada classe (**e.g., .destaque**).

ID: Selecionam um elemento único com um ID específico (**e.g., #cabecalho**).

Descendentes: Selecionam elementos que estão dentro de outro elemento (**e.g., div p**).

Adjacentes: Selecionam elementos que são irmãos e vêm imediatamente após um elemento específico (**e.g., h1 + p**).

PROPRIEDADES CSS

As propriedades CSS são os **atributos** que podem ser aplicados aos elementos, como cor, fonte, tamanho, espaçamento, etc. Algumas das propriedades mais comuns incluem:

color: Define a cor do texto.

```
p {  
    color: blue;  
}
```

Font-size: Define o tamanho da fonte.

```
h1 {  
    font-size: 36px;  
}
```

Background-color: Define a cor de fundo.

```
div {  
    background-color: lightgray;  
}
```

Width: Define a largura de um elemento.

```
img {  
    width: 200px;  
}
```

PROPRIEDADES CSS

Height: Define a altura de um elemento.

```
div {  
    height: 150px;  
}
```

Margin: Define o espaço ao redor de um elemento.

```
p {  
    margin: 20px;  
}
```

Padding: Define o espaço interno de um elemento.

```
div {  
    padding: 10px;  
}
```

LINKANDO O CSS AO HTML

Existem três principais formas de ligar um arquivo CSS ao seu documento HTML:

Link externo:

HTML

```
<link rel="stylesheet" href="styles.css">
```

Estilo interno:

HTML

```
<style>
  /* Estilos aqui */
</style>
```

Estilo inline:

HTML

```
<p style="color: red;">Este texto é vermelho.</p>
```

Sugestão para aprimorar seus conhecimentos em CSS:



Link: <https://youtu.be/AB35iSr1YyA?si=E8o3XRBqFIPLJ-sL>

PRATIQUE E APLIQUE OS CONHECIMENTOS ADQUIRIDOS NESTE MÓDULO.

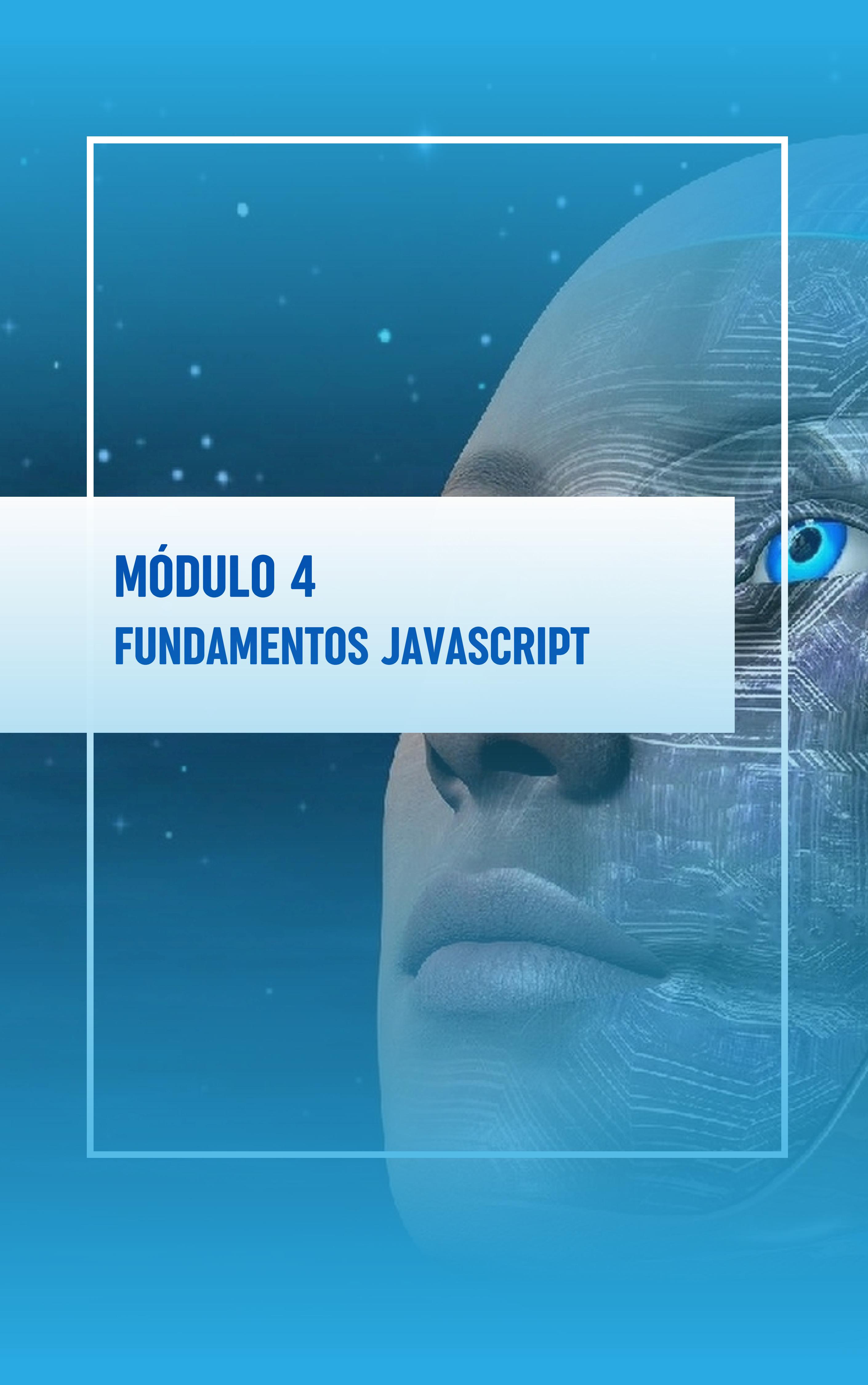
1 - Qual das alternativas abaixo é uma vantagem do uso de CSS em páginas web?

- a) Permite a inserção de scripts dinâmicos no site.
- b) Facilita a manutenção do código separando conteúdo e apresentação.
- c) Garante a segurança da página contra ataques cibernéticos.
- d) Gera automaticamente o layout para diferentes tamanhos de tela.

2 - Qual dos seletores CSS é utilizado para estilizar um elemento único com um ID específico?

- a) Seletor de tipo
- b) Seletor de classe
- c) Seletor de ID
- d) Seletor descendente

3 - Explique a importância do CSS na construção de páginas web e descreva ao menos dois tipos de seletores utilizados para estilizar elementos HTML.



MÓDULO 4

FUNDAMENTOS JAVASCRIPT

OQUE É JAVASCRIPT?

JavaScript é uma **linguagem de programação** utilizada por desenvolvedores para criar páginas interativas na Internet.

Ela melhora a experiência do usuário em sites, permitindo atualizações dinâmicas, animações e recursos interativos, como carrosséis de imagens e menus suspensos.

Como uma linguagem de script do **lado do cliente**, JavaScript é fundamental para a funcionalidade da web moderna.

Por que usar JavaScript?

Interatividade: JavaScript permite que as páginas da Web respondam a ações do usuário, como cliques e movimentações do mouse, tornando a experiência mais envolvente.

Dinamismo: Com JavaScript, o conteúdo da página pode ser atualizado sem a necessidade de recarregar, permitindo a criação de aplicativos Web interativos, como redes sociais e plataformas de e-commerce.

Manipulação do DOM: A linguagem possibilita a modificação do Document Object Model (DOM), permitindo que elementos da página sejam alterados, adicionados ou removidos em tempo real.

COMO O JAVASCRIPT FUNCIONA?

O JavaScript é categorizado como uma **linguagem de criação de scripts** ou uma linguagem interpretada. O código em JavaScript é **interpretado**, ou seja, é traduzido diretamente no código de linguagem de máquina subjacente por um mecanismo de Javascript.

Em outras linguagens de programação, um compilador compila todo o código em linguagem de máquina durante uma etapa distinta.

Portanto, todas as linguagens de criação de script são linguagens de programação, mas nem todas as linguagens de programação são linguagens de criação de script.

Mecanismo JavaScript:

Um mecanismo JavaScript é um **programa de computador** que executa código em JavaScript.

Os primeiros mecanismos eram simples interpretadores, mas todos os mecanismos modernos usam compilação just-in-time ou em tempo de execução para melhorar a performance.

SINTAXE BÁSICA EM JAVASCRIPT

A sintaxe básica é composta de vários **elementos fundamentais** que facilitam a criação de scripts interativos para páginas web e outros aplicativos. Aqui estão alguns dos principais elementos:

Variáveis

Variáveis são usadas para **armazenar dados** em JavaScript. Vamos ver como declarar variáveis e atribuir valores a elas:

```
// Declarando variáveis
var nome = "João";
let idade = 30;
const PI = 3.14;

// Reatribuindo valores
nome = "Maria";
idade = 25;
// PI = 3.1415; // Isso geraria um erro, pois PI é uma constante
```

Vetores

Vetores (ou arrays) são usados para **armazenar coleções de valores**. Eles são declarados usando colchetes '[]'. Veja um exemplo de como criar e acessar elementos em um vetor:

```
// Declarando um vetor
let frutas = ["maçã", "banana", "laranja"];

// Acessando elementos
console.log(frutas[0]); // Saída: maçã
console.log(frutas.length); // Saída: 3
```

SINTAXE BÁSICA EM JAVASCRIPT

Objetos

Objetos são estruturas de dados que armazenam pares chave-valor. Eles são declarados usando chaves '{ }'. Aqui está um exemplo de um objeto que representa uma pessoa:

```
// Declarando um objeto
let pessoa = {
  nome: "Ana",
  idade: 28,
  cidade: "São Paulo"
};

// Acessando propriedades
console.log(pessoa.nome); // Saída: Ana
console.log(pessoa["idade"]); // Saída: 28
```

Estruturas Condicionais

Estruturas condicionais permitem que você **tome decisões** em seu código com base em condições. O 'if', 'else' if 'e 'else' são frequentemente usados para isso:

```
let idade = 18;

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

SINTAXE BÁSICA EM JAVASCRIPT

Funções

Funções são blocos de código reutilizáveis que podem receber argumentos e retornar valores. Aqui está um exemplo simples de uma função que calcula a soma de dois números:

```
// Declarando uma função
function somar(a, b) {
  return a + b;
}

// Chamando a função
let resultado = somar(5, 3);
console.log(resultado); // Saída: 8
```

As funções também podem ser **armazenadas em variáveis** e passadas como argumentos para outras funções, tornando-as extremamente flexíveis.

Resumo

Em suma, a sintaxe básica em JavaScript envolve a declaração de variáveis, manipulação de vetores e objetos, uso de estruturas condicionais para tomar decisões e a criação de funções para organizar e reutilizar código. Esses conceitos são **fundamentais** para dominar JavaScript e criar aplicações web dinâmicas e interativas.

MANIPULAÇÃO DO DOM

A manipulação do DOM (Document Object Model) é um dos principais recursos do JavaScript para **interagir e modificar** o conteúdo e a estrutura de páginas web.

O DOM é uma **representação da estrutura HTML** de uma página como um **conjunto de objetos**, permitindo que JavaScript acesse e altere elementos, atributos, estilos e até mesmo crie novos elementos dinamicamente.

O que é o DOM?

- O DOM é uma **interface de programação** que representa o conteúdo HTML como uma **árvore de nós**, onde cada nó corresponde a uma parte do documento (tags, texto, atributos, etc).
- Com o DOM, o JavaScript pode **modificar** os elementos da página, **adicionar** novos elementos ou **remover** os existentes, além de **reagir** a eventos.

Principais formas de manipular o DOM com JavaScript:

Selecionando Elementos

Para manipular um elemento, primeiro você precisa **selecioná-lo**. JavaScript fornece várias formas de acessar elementos HTML.

MANIPULAÇÃO DO DOM

getElementById()

Seleciona um elemento pelo seu ID:

```
let elemento = document.getElementById("minhaDiv");
console.log(elemento); // Exibe o elemento com o ID "minhaDiv"
```

getElementsByClassName()

Seleciona todos os elementos que possuem uma determinada classe:

```
let elementos = document.getElementsByClassName("minhaClasse");
console.log(elementos); // Exibe uma coleção de todos os elementos com a classe "minhaClasse"
```

Modificando o Conteúdo dos Elementos

Depois de selecionar um elemento, você pode alterar seu conteúdo:

innerHTML

Altera ou obtém o conteúdo HTML de um elemento:

```
let div = document.getElementById("minhaDiv");
div.innerHTML = "<p>Novo conteúdo</p>"; // Substitui o conteúdo da div por um novo parágrafo
```

MANIPULAÇÃO DO DOM

Modificando Atributos

Você também pode **alterar** os atributos de um elemento HTML, como classes, ids, links, etc.

setAttribute()

Define o valor de um atributo específico:

```
let link: HTMLAnchorElement | null = ...;
link.setAttribute("href", "https://exemplo.com"); // Muda o destino do link
```

Resumo

O DOM (Document Object Model) permite que o JavaScript **interaja diretamente** com o HTML, possibilitando uma vasta gama de **manipulações** em uma página web. Aqui está um resumo do que pode ser feito com o DOM:

- **Selecionar elementos:** Acessar elementos HTML usando métodos como `getElementById()`, `getElementsByClassName()`, `querySelector()`, entre outros.

MANIPULAÇÃO DO DOM

- **Modificar conteúdo:** Alterar o texto ou HTML interno dos elementos com innerHTML ou textContent.
- **Modificar atributos:** Adicionar, remover ou alterar atributos como src, href, class, etc., usando setAttribute(), getAttribute() e removeAttribute().
- **Modificar estilos (CSS):** Alterar diretamente os estilos CSS dos elementos com a propriedade style.
- **Adicionar e remover elementos:** Criar novos elementos com createElement() e adicioná-los à página com appendChild(), ou remover elementos com remove() ou removeChild().
- **Reagir a eventos:** Detectar e responder a interações do usuário, como cliques, teclas pressionadas, e muito mais, com addEventListener().

Essas funcionalidades permitem criar páginas dinâmicas, interativas e com conteúdo personalizado, respondendo às ações do usuário em tempo real.

PRATIQUE E APLIQUE OS CONHECIMENTOS ADQUIRIDOS NESTE MÓDULO.

1 - Qual das alternativas abaixo descreve corretamente o que é o Document Object Model (DOM) no contexto do JavaScript?

- a) Um modelo de banco de dados usado para armazenar grandes quantidades de informações dinâmicas.
- b) Uma interface de programação que representa a estrutura HTML de uma página como uma árvore de nós.
- c) Um método para organizar o código JavaScript em diferentes módulos.
- d) Um framework JavaScript para criar aplicações web interativas.

2 - Qual método JavaScript é utilizado para selecionar um elemento HTML pelo seu ID?

- a) querySelector()
- b) getElementsByClassName()
- c) getElementById()
- d) innerHTML()

3 - Como o JavaScript transforma páginas da Web de estáticas para dinâmicas?

GABARITO - INTERNET

Internet -

1 - Como uma página web é geralmente dividida em termos de estrutura e funcionamento?

Resposta: A) Frontend e Backend

2 - Qual das alternativas não descreve corretamente a função de um componente da rede na Internet?

Resposta: D) Switches conectam redes locais diferentes, permitindo a comunicação entre elas.

3 - Explique o papel dos roteadores e switches na comunicação de dados em redes, destacando as diferenças entre suas funções.

Possível resposta:

Roteadores conectam diferentes redes, gerenciando o tráfego entre elas e direcionando pacotes de dados pelo caminho mais eficiente. Eles permitem a comunicação entre redes locais (LAN) e externas, como a Internet. Já os switches conectam dispositivos dentro de uma mesma rede local (LAN), permitindo a troca de dados entre eles de forma eficiente, sem sair da rede.

GABARITO - HTML

HTML -

1 - Qual das seguintes tags HTML é usada para definir o título principal de uma página web?

Resposta: B) <h1>

2 - Qual é a função da tag <nav> em HTML?

Resposta: C) Criar uma seção de navegação

3 - Explique a importância de uma estrutura semântica no HTML e cite dois elementos semânticos que ajudam na acessibilidade e SEO de uma página.

Possível resposta:

Uma estrutura semântica no HTML é importante porque ajuda os navegadores e motores de busca a entender melhor o conteúdo da página, facilitando a indexação correta para SEO (Search Engine Optimization) e melhorando a acessibilidade para leitores de tela e outras tecnologias assistivas.

Dois exemplos de elementos semânticos são:

<header>: Define o cabeçalho de uma página ou seção, geralmente contendo logotipos, títulos e menus de navegação.

<main>: Representa o conteúdo principal da página, ajudando a identificar a parte mais relevante para os usuários e motores de busca.

GABARITO - CSS

CSS -

1 - Qual das alternativas abaixo é uma vantagem do uso de CSS em páginas web?

Resposta: B) Facilita a manutenção do código separando conteúdo e apresentação.

2 - Qual dos seletores CSS é utilizado para estilizar um elemento único com um ID específico?

Resposta: C) Seletor de ID

3 - Explique a importância do CSS na construção de páginas web e descreva ao menos dois tipos de seletores utilizados para estilizar elementos HTML.

Possível resposta:

O CSS é essencial porque permite separar a estrutura do conteúdo (HTML) da aparência, facilitando a criação de layouts esteticamente agradáveis. Além disso, ele possibilita a criação de sites responsivos, que se adaptam a diferentes dispositivos e tamanhos de tela.

Dois tipos de seletores comumente usados no CSS são:

Seletor de classe: Utilizado para estilizar múltiplos elementos que compartilham uma mesma classe. Ele é precedido por um ponto, por exemplo: .classeExemplo.

Seletor de ID: Usado para estilizar um único elemento com um identificador único. Ele é precedido pelo símbolo #, por exemplo: #idExemplo.

GABARITO - JAVASCRIPT

JavaScript -

1 - Qual das alternativas abaixo descreve corretamente o que é o Document Object Model (DOM) no contexto do JavaScript?

Resposta: B) Uma interface de programação que representa a estrutura HTML de uma página como uma árvore de nós.

2 - Qual método JavaScript é utilizado para selecionar um elemento HTML pelo seu ID?

Resposta: C) getElementById()

3 - Como o JavaScript transforma páginas da Web de estáticas para dinâmicas?

Resposta: O JavaScript transforma páginas da Web de estáticas para dinâmicas ao permitir a manipulação do DOM, o que possibilita atualizações de conteúdo em tempo real sem recarregar a página.

REFERÊNCIAS

HOSTINGER. O que é HTML? Conceitos básicos. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-html-conceitos-basicos>. Acesso em: 16 out. 2024.

MOZILLA DEVELOPER NETWORK. Getting started – Introdução ao HTML. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/Getting_started. Acesso em: 16 out. 2024

MOZILLA DEVELOPER NETWORK. HTML basics – Fundamentos do HTML. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/Getting_started_with_the_web/HTML_basics. Acesso em: 16 out. 2024.

SEMRUSH. Tags HTML. Disponível em: <https://pt.semrush.com/blog/tags-html/>. Acesso em: 16 out. 2024.

HOMEHOST. Tags HTML: O que são e como utilizá-las. Disponível em: <https://www.homehost.com.br/blog/tutoriais/tags-html/>. Acesso em: 16 out. 2024.

W3SCHOOLS. HTML Elements. Disponível em: https://www.w3schools.com/html/html_elements.asp. Acesso em: 16 out. 2024.

TABLELESS. Estrutura Básica de uma Página HTML. Disponível em: <https://tableless.github.io/iniciantes/manual/html/estruturabasica.html>. Acesso em: 16 out. 2024

HIPERBYTES. Curso de HTML – Aula 1: Estrutura básica de páginas web. Disponível em: <https://www.hiperbytes.com.br/curso-de-html-aula1-estrutura-basica-paginas-web/>. Acesso em: 16 out. 2024.

Hostinger. O que é CSS: Guia básico de CSS. Recuperado de <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css>. Acesso em: 19 out. 2024

Mozilla Developer Network (MDN). CSS básico. Recuperado de https://developer.mozilla.org/pt-BR/docs/Learn/Getting_started_with_the_web/CSS_basics. Acesso em: 19 out. 2024

REFERÊNCIAS

Mozilla Developer Network (MDN). O que é CSS. Recuperado de https://developer.mozilla.org/pt-BR/docs/Learn/CSS/First_steps/What_is_CSS. Acesso em: 19 out. 2024

DIO. O que é CSS. Recuperado de <https://www.dio.me/articles/o-que-e-css-NO9GM4>. Acesso em: 19 out. 2024

Canal do YouTube. (2024, 19 de outubro). O QUE É CSS? (SELETORES, PROPRIEDADES E VALORES) YouTube. <https://www.youtube.com/watch?v=LWU2OR19ZG4>. Acesso em: 19 out. 2024