1. Explique a diferença entre uma linguagem de programação compilada e uma linguagem interpretada, e identifique em qual dessas categorias o JavaScript se encaixa.

R=

* **Compiladas**: O código é convertido em binário antes da execução (ex: C, C++).
* **Interpretadas**: O código é executado linha por linha em tempo real (ex: JavaScript, Python).

JavaScript é considerado uma linguagem interpretada, embora em alguns contextos utilize técnicas de compilação em tempo de execução.

2) Como o JavaScript evoluiu desde sua criação? Mencione pelo menos três áreas ou plataformas em que a linguagem é utilizada hoje em dia.

R=

O JavaScript evoluiu de uma linguagem simples para uma poderosa ferramenta de desenvolvimento, utilizada em diversas áreas, como:

* Desenvolvimento web front-end.
* Aplicações móveis multiplataforma.
* Desenvolvimento de servidores e APIs.

Essa evolução contínua e a versatilidade da linguagem contribuíram para sua popularidade e adoção em uma ampla variedade de contextos tecnológicos.

3) Descreva o papel dos verbos HTTP em uma aplicação web. Escolha dois verbos HTTP (por exemplo, GET e POST) e explique suas funcionalidades e quando devem ser utilizados.

R=

* **GET**:
  + **Funcionalidade**: Solicita dados.
  + **Uso**: Recuperação de informações sem alterar o estado do recurso.
* **POST**:
  + **Funcionalidade**: Envia dados ao servidor.
  + **Uso**: Criação ou atualização de recursos.

Compreender a funcionalidade e a aplicação dos verbos HTTP é essencial para projetar APIs e aplicações web de forma eficiente e semântica.

4) Explique o que é uma API RESTful e como os verbos HTTP se integram ao desenvolvimento de APIs.

R= Uma API RESTful é uma interface que permite a comunicação entre sistemas utilizando recursos acessíveis via URLs e verbos HTTP. Os verbos HTTP são essenciais para definir as operações que podem ser realizadas nos recursos, proporcionando uma estrutura clara e semântica para a interação entre clientes e servidores. Essa abordagem é amplamente adotada devido à sua simplicidade, flexibilidade e capacidade de escalar.

5) Node.js é amplamente utilizado para desenvolvimento no backend. Quais são as principais vantagens de usar o Node.js no lado do servidor em comparação a outras linguagens como Java ou Python?

R= Node.js oferece vantagens significativas em termos de desempenho, escalabilidade, unificação do desenvolvimento com JavaScript, um ecossistema rico e suporte para aplicações em tempo real. Essas características o tornam uma escolha popular para uma variedade de aplicações, especialmente aquelas que exigem alta concorrência e desenvolvimento ágil.

6) Discuta o conceito de "modelo de I/O não bloqueante" no Node.js e explique por que ele é vantajoso para aplicações em tempo real.

R= O modelo de I/O não bloqueante do Node.js permite que operações de entrada/saída sejam gerenciadas de maneira assíncrona, resultando em alta concorrência, eficiência de recursos, baixa latência e facilidade de escalabilidade. Essas características tornam o Node.js uma escolha excelente para construir aplicações em tempo real, onde a rapidez e a capacidade de lidar com múltiplas conexões são cruciais para a experiência do usuário.

7) JavaScript suporta múltiplos paradigmas de programação. Descreva os três principais paradigmas suportados pela linguagem e forneça um exemplo de uso para cada um.

R= JavaScript suporta:

1. **Programação Imperativa**: Enfoca a execução sequencial de comandos (exemplo: uso de loops).

let total = 0;

for (let i = 1; i <= 5; i++) {

total += i; // Adiciona i ao total em cada iteração

}

console.log(total); // Saída: 15

1. **Programação Funcional**: Utiliza funções como entidades de primeira classe (exemplo: uso de map).

const numbers = [1, 2, 3, 4, 5];

// Usando map para criar um novo array com o quadrado dos números

const squares = numbers.map(num => num \* num);

console.log(squares); // Saída: [1, 4, 9, 16, 25]

1. **Programação Orientada a Objetos**: Organiza código em objetos e classes (exemplo: criação de um objeto Car).

// Definindo um objeto com um construtor

function Car(make, model) {

this.make = make;

this.model = model;

}

// Adicionando um método ao protótipo

Car.prototype.displayInfo = function() {

return `Carro: ${this.make} ${this.model}`;

};

// Criando uma instância do objeto Car

const myCar = new Car('Toyota', 'Corolla');

console.log(myCar.displayInfo()); // Saída: Carro: Toyota Corolla

Esses paradigmas oferecem flexibilidade aos desenvolvedores, permitindo que eles escolham o estilo que melhor se adapta às suas necessidades e ao problema em questão.

8) Qual a importância do npm (Node Package Manager) no desenvolvimento com Node.js? Descreva como ele facilita o gerenciamento de dependências em projetos.

R= O npm é uma ferramenta essencial para o desenvolvimento com Node.js, pois:

* Facilita o gerenciamento de dependências, permitindo a instalação, atualização e remoção de pacotes de maneira simples.
* Garante a compatibilidade de versões através do arquivo package.json.
* Permite a automação de tarefas com scripts personalizados.
* Oferece um vasto ecossistema de pacotes, promovendo a reutilização de código.

Essas funcionalidades tornam o npm uma parte integral do fluxo de trabalho de desenvolvimento em Node.js, melhorando a eficiência e a organização dos projetos.

9) Quais são as principais características de uma linguagem de programação de alto nível? Cite exemplos de como o JavaScript implementa essas características.

R= As principais características de uma linguagem de programação de alto nível incluem:

* **Abstração**: Oculta os detalhes do hardware.
* **Sintaxe Legível**: Facilita a leitura e a escrita do código.
* **Gerenciamento Automático de Memória**: Coleta de lixo para gerenciar a memória.
* **Portabilidade**: O mesmo código pode ser executado em diferentes plataformas.
* **Suporte a Estruturas de Dados Complexas**: Facilita o gerenciamento de dados.
* **Interpretação e Compilação**: Flexibilidade na execução do código.

O JavaScript implementa essas características de forma eficaz, tornando-o uma linguagem poderosa e popular para o desenvolvimento web e além.

10) Embora o JavaScript seja essencial para adicionar interatividade a websites, HTML e CSS desempenham papéis cruciais no desenvolvimento web. Explique a função de cada uma dessas três tecnologias na criação de páginas web.

* **HTML**: Estrutura o conteúdo da página, definindo os elementos e sua semântica.
* **CSS**: Estiliza a apresentação visual, controlando como os elementos aparecem e se comportam.
* **JavaScript**: Adiciona interatividade e dinamismo, permitindo a manipulação do conteúdo e a resposta a ações do usuário.

Essas três tecnologias trabalham em conjunto para criar experiências web ricas, onde o HTML fornece a base, o CSS dá estilo e o JavaScript traz vida à página.