

1. Explique o que é o jsFiddle e qual a sua utilidade no desenvolvimento com Vue.js

O jsFiddle é uma ferramenta online para testar, editar e compartilhar códigos HTML, CSS e JavaScript.

No desenvolvimento com Vue.js, ele é útil para criar protótipos rápidos, testar componentes e demonstrar problemas ou soluções sem precisar configurar um projeto local.

2. Descreva o processo para configurar o Vue.js no jsFiddle. Quais os passos necessários para incluir a biblioteca Vue.js?

1. Acesse o site jsfiddle.net.
2. No menu superior, clique em **Settings** (ícone da engrenagem).
3. Vá em **External Resources**.
4. Adicione o link CDN do Vue.js, como:
https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js (Vue 2) ou
https://cdn.jsdelivr.net/npm/vue@3/dist/vue.global.js (Vue 3).
5. Clique em **Update** para aplicar.
6. Escreva o HTML, JavaScript e CSS normalmente e clique em **Run** para ver o resultado.

3. O que é o "Hello World" no Vue.js e qual a sua importância no aprendizado do framework?

O "Hello World" é o exemplo mais simples de uso do Vue.js, onde se exibe uma mensagem na tela com o framework.

É importante porque mostra a estrutura básica de um componente Vue, a ligação entre HTML e dados, e como o Vue reativa a interface. É o primeiro passo para entender a lógica do framework.

4. Defina o conceito de two-way data binding no Vue.js. Por que ele é uma característica importante?

Two-way data binding é a ligação automática entre o modelo (dados) e a interface (HTML), de forma que mudanças em um afetam o outro instantaneamente.

É importante porque facilita a sincronização entre o que o usuário vê e o que é

processado no código, tornando o desenvolvimento de formulários e interfaces interativas mais simples e eficiente.

5. Qual a principal função do v-for no Vue.js? Dê um exemplo de quando ele é utilizado.

O v-for serve para **criar listas** dinâmicas de elementos com base em arrays ou objetos.

```
<li v-for="nome in nomes">{{ nome }}</li>
```

6. Explique o propósito da diretiva v-bind:key ao trabalhar com listas dinâmicas no Vue.js.

A diretiva v-bind:key serve para **identificar de forma única** cada item de uma lista renderizada com v-for.

Isso ajuda o Vue a rastrear e **atualizar os elementos corretamente**, melhorando a performance e evitando erros na renderização dinâmica.

7. O que é a reatividade no Vue.js e como ela simplifica o desenvolvimento de interfaces dinâmicas?

Reatividade no Vue.js é a capacidade do framework de **detectar mudanças nos dados** e atualizar automaticamente a interface.

Isso simplifica o desenvolvimento, pois o desenvolvedor não precisa manipular o DOM manualmente — o Vue cuida das atualizações visuais sempre que os dados mudam.

8. Quais são as principais vantagens de usar o ciclo de vida dos componentes no Vue.js? Cite pelo menos dois hooks e suas finalidades.

O ciclo de vida permite **executar ações em momentos específicos** da criação, atualização e destruição do componente.

Vantagens: melhor controle do comportamento, carregamento de dados no momento certo e limpeza de recursos.

Hooks:

- `mounted()`: executado após o componente ser inserido no DOM (usado para chamadas à API).
- `destroyed()`: executado antes do componente ser removido (usado para limpar timers ou eventos).

9. Diferencie os métodos JavaScript push e splice ao manipular arrays no Vue.js. Como o Vue detecta alterações feitas com esses métodos?

- push() adiciona um ou mais itens ao final do array.
- splice() pode adicionar, remover ou substituir itens em qualquer posição do array.

O Vue **detecta alterações** feitas com esses métodos porque eles são **reativos**, ou seja, modificam o array de forma que o Vue consegue observar e refletir essas mudanças na interface automaticamente.

10. O que é o método Vue.set e em quais cenários ele é necessário para garantir a reatividade?

O Vue.set é usado para **adicionar novas propriedades reativas** a objetos ou arrays que já existem.

É necessário quando se adiciona dinamicamente uma chave a um objeto, pois o Vue não detecta novas propriedades criadas diretamente.

11. Explique o uso dos modificadores de evento .prevent e .stop no Vue.js. Como eles afetam o comportamento de um evento do DOM?

- .prevent chama event.preventDefault(), impedindo o comportamento padrão do evento (ex: envio de formulário).
- .stop chama event.stopPropagation(), impedindo que o evento se propague para elementos-pai.

Ambos modificam o comportamento do DOM para dar mais controle ao desenvolvedor sobre a interação do usuário.

12. Descreva como os modificadores de teclas, como .enter e .esc, são usados no Vue.js. Cite um exemplo prático de aplicação.

Modificadores como .enter e .esc permitem executar ações somente quando teclas específicas são pressionadas.

```
<input @keyup.enter="enviarMensagem" />
```

13. Quais são as diferenças entre checkboxes, radios e selects em formulários dinâmicos no Vue.js?

- **Checkboxes** permitem múltiplas seleções e vinculam-se a arrays.
- **Radios** permitem apenas uma seleção entre várias opções e se ligam a um valor único.
- **Selects** podem funcionar como radios (modo simples) ou checkboxes (com multiple).

No Vue.js, todos podem ser ligados com v-model para atualizar os dados automaticamente conforme a interação do usuário.

14. Por que é importante usar o atributo v-model ao criar formulários no Vue.js?

Porque o v-model cria uma ligação **bidirecional** entre o campo do formulário e o dado no componente.

Isso mantém os dados atualizados automaticamente com o valor digitado pelo usuário, facilitando a manipulação e validação de formulários de forma simples e eficiente.