

Teil A: Grundverständnisfragen

1. Welche grundlegenden Bestandteile enthält ein HTML-Dokument?

Antwort: Ein HTML-Dokument besteht aus dem `<html>` Tag, welches den gesamten Inhalt des Dokuments umschließt. Innerhalb des `<html>` Tags befinden sich der `<head>` und der `<body>` Tag. Der `<head>` Tag enthält Metadaten und Verweise auf externe Ressourcen, während der `<body>` Tag den sichtbaren Inhalt der Webseite enthält.

2. Was ist der Zweck eines HTML-Tags und wie wird er definiert?

Antwort: Ein HTML-Tag wird verwendet, um den strukturierten Inhalt einer Webseite zu definieren. Ein Tag besteht aus einem Namen, der von spitzen Klammern umschlossen ist, z.B. `<tagname>`. Einige Tags können auch Attribute enthalten, die zusätzliche Informationen über den Tag angeben, z.B. `<tagname attribute="value">`.

3. Nenne drei Beispiele für HTML Tags.

Antwort: Einige Beispiele für HTML-Tags sind `<h1>` für Überschriften, `<p>` für Absätze und `` für Bilder.

4. Nennen Sie zwei verschiedene primitive Datentypen in JavaScript und beschreiben Sie diese kurz.

Antwort: Boolean: Boolean ist ein Wahrheitswert. Es gibt nur die Werte "wahr" oder "falsch" (binär). String: Ein String ist eine Aneinanderkettung von Zeichen.

5. Welche drei Blöcke kann ein Single File Component (SFC) in Vue enthalten?

Antwort: `<template>` für das HTML Template, `<script>` für JavaScript Code und `<style>` für CSS

6. Was kann mit der Vue Direktive v-on erreicht werden?

Antwort: Mit dieser Direktive kann eine Handlerfunktion für ein DOM Event definiert werden (z.B. `@click(...)`).

7. Erklären Sie den Unterschied zwischen CSS-Selektoren der Klasse und der ID.

Antwort: Klassen-Selektoren (`.classname`) werden verwendet, um mehreren Elementen denselben Stil zuzuweisen. ID-Selektoren (`#idname`) sind einzigartig und stylen ein spezifisches Element.

Teil B: Multiple-Choice-Fragen

Hinweis: Es gibt immer nur 1 richtige Antwort.

Frage 1

Welches Schlüsselwort wird verwendet, um eine Variable in JavaScript zu deklarieren, die nur innerhalb eines bestimmten Blocks sichtbar ist?

- A) `global`
- B) `private`
- C) `var`
- D) `let`

Richtige Antwort: D) `let`

Frage 2

Welches Zeichen wird in JavaScript verwendet, um eine Zeichenkette zu verketten?

- A) `+`
- B) `&`
- C) `*`
- D) `-`

Richtige Antwort: A) `+`

Frage 3

Welches Ergebnis gibt `5 == '5'` in JavaScript zurück?

- A) `true`
- B) `false`
- C) `NaN`

D) undefined

Richtige Antwort: A) true

Teil C: Codebeispiele und Verständnisfragen

Vue.js-Beispiel mit Composition API

Gegeben sei folgender Vue.js-Code:

```
<div id="app">
  {{ message }}
</div>

<script>
import { ref } from 'vue';

createApp({
  setup() {
    const message = ref('Hallo Vue!');
    return { message };
  }
}).mount('#app');
</script>
```

Fragen:

1. Was zeigt dieses Vue.js-Beispiel auf der Webseite an?
2. Wie würden Sie die `message`-Eigenschaft ändern, um den Text 'Willkommen bei Vue.js' anzuzeigen?

Lösungen:

1. Das Beispiel zeigt den Text 'Hallo Vue!' auf der Webseite an.
2. Um die `message`-Eigenschaft zu ändern, ersetzen Sie den Wert von `message` innerhalb der `setup`-Funktion durch `const message = ref('Willkommen bei Vue.js');`.

Vue.js-Beispiel 2

Betrachten Sie folgenden Code:

```

<template>
  <div>
    <span>{{ doubled }}</span>
  </div>
</template>

<script setup>
import { ref, computed } from 'vue';

const num = ref(5);
const doubled = computed(() => num.value * 2);
</script>

```

Frage:

- Was wird durch die Ausführung dieses Codes erreicht? Beschreibe kurz den Ablauf.

Lösung:

- Der Wert der reaktiven Variablen "num" wird auf 5 gesetzt. Das computed Property "doubled" nimmt den Wert dieser reaktiven Variable und multipliziert ihn mit 2. Ein Computed Property aktualisiert sich immer selbst, wenn ein Wert, der in diesem verwendet wird, sich ändert. Daher wird im Tag 10 gerendert.

Vue.js-Beispiel 3

Betrachten Sie folgenden Code:

```

<template>
  <div>
    <button @click="toggle">Toggle</button>
    <p v-if="visible">Content is visible</p>
  </div>
</template>

<script setup>

```

```
import { ref } from 'vue';

const visible = ref(false);

const toggle = () => {
  visible.value = !visible.value;
};
</script>
```

Frage:

1. Was wird durch die Ausführung dieses Codes erreicht? Beschreibe kurz den Ablauf.

Lösung:

1. Die Variable "visible" ist reactive und beinhaltet einen Wahrheitswert, der auf false gesetzt wird. Mit der v-if Direktive wird ein <p> Tag gerendert, das gerendert werden soll, wenn die Variable "visible" auf true steht. Mit der Pfeilfunktion "toggle" wird der Wert der Variable "visible" invertiert. Diese Funktion wird aufgerufen, wenn der Button im Template geklickt wird. Dies wird durch die v-on Direktive erreicht. Wenn der Knopf also gedrückt wird, wird das <p> Tag gerendert oder nicht gerendert.