

# Modul - Kommunikation zwischen Client und Server

\* Dieses Formular wird Ihren Namen aufzeichnen. Bitte tragen Sie Ihren Namen ein.

1

Welchen Aussage(n) zu **let** und **const** in JavaScript sind korrekt? (1 Punkt)

- ☐ Variablen mit const sind neu zuweisbar und nicht neu deklarierbar.
- ☐ Variablen mit let sind neu zuweisbar und nicht neu deklarierbar.
- ☐ Variablen mit const sind nicht neu zuweisbar und nicht neu deklarierbar.

2

Im Unterschied zu **var** haben **let** und **const** ... (1 Punkt)

- ☐ ... eine maximale 8 Byte Speichergröße. Daher können keine längere Zeichenketten gespeichert werden.
- ☐ ... einen Blockbereich. Das heißt, wenn Sie sie innerhalb eines Blocks deklarieren, sind sie nur innerhalb dieses Blocks zugänglich {}.
- ☐ ... kaum eine Verwendung in JavaScript.

Welchen Output erzeugt das gezeigte Code-Beispiel? (1 Punkt)

```
let x = 19.00 + "Hallo 12. Klasse" + 4;  
console.log(x);
```

- ☐ 19.00
- ☐ Hallo 12. Klasse
- ☐ Hallo 12. Klasse4
- ☐ 19Hallo 12. Klasse4
- ☐ 19.00Hallo 12. Klasse4

Lassen Sie sich den Namen des Schülers ausgeben:  
**"Der Schüler heisst Max Schmied"** (1 Punkt)

```
const student = {  
  name: "Max Schmied",  
  klasse: "12Fix",  
  alter: "19"  
};
```

- ☐ print("Der Schüler heisst \${student.name}");
- ☐ console.log(`Der Schüler heisst \${student.name}`);
- ☐ console.log("Der Schüler heisst \${student.name}");
- ☐ print(`Der Schüler heisst \${student.name}`);

5

Die Funktion ***findDuplicates()*** soll alle doppelten Zahlen ausgeben. Leider ist die Funktion Fehlerhaft. Finden Sie den Fehler.

```
1  function findDuplicates() {  
2      const numbers = [2, 4, 5, 2, 5, 8, 5, 10];  
3      for(let i=0; i<numbers.length; i++){  
4          const numberAtI = numbers[i];  
5          for(let j=0; j<numbers.length; j++) {  
6              const numberAtJ = numbers[j];  
7              if(numberAtI === numberAtJ) {  
8                  console.log(`Doppelte Zahl gefunden:${numberAtI}`);  
9              }  
10         }  
11     }  
12 }
```

6

Was ist der Unterschied zwischen der asynchronen und synchronen Kommunikation zwischen Client und Server?

7

Welche Vorteile hat die asynchrone Kommunikation im Gegensatz zur synchronen Kommunikation?

- ☐ bessere Übertragungszeit
- ☐ weniger virtueller Speicher
- ☐ Interaktion mit Komponenten während andere Komponenten noch laden

8

Was ist eine Callback-Hell? (1 Punkt)

- ☐ Vermehrte Verschachtelung von asynchronen Funktionsaufrufen in JavaScript.
- ☐ Code-Fehler durch schlechte JavaScript-Entwickler
- ☐ Synchrone Funktionen innerhalb einer asynchronen Funktion, welches wiederum in einer synchronen Funktion ist.

Betrachten Sie folgendes Promise. Bringen Sie die Code-Schnipsel unten in die richtige Reihenfolge für einen sinnvollen Output. (1 Punkt)

```
const promise = new Promise(function (resolve, reject) {  
  const x = "12 Klasse ist super";  
  const y = "12 Klasse ist super";  
  if (x === y) {  
    resolve();  
  } else {  
    reject();  
  }  
});
```

promise

.then(function () {

console.log("Promise war erfolgreich!");

})

.catch(function() {

console.log("Promise wurde zurückgewiesen!");

});

### Was machen Async/Await? (1 Punkt)

- ☐ Sie sind "syntaktischer Zucker" für Promises und sind ein Wrapper um Code umzugestalten. Sie machen das lesen und verwenden von Code einfacher.
- ☐ Sie beseitigen die "Pyramid of Doom" bei Promises.
- ☐ Intern wird bei einer async markierten Funktion immer ein Promise-Objekt erzeugt(ausdrücklich oder automatisch falls nicht ausdrücklich angegeben) und zurückgegeben. Auf das Ergebnis kann über die direkte Verwendung von Promises zurückgegriffen werden.

---

Dieser Inhalt wurde von Microsoft weder erstellt noch gebilligt. Die von Ihnen übermittelten Daten werden an den Formulareigentümer gesendet.



Microsoft Forms