

# JavaScript Aufgaben: Funktionen, Schleifen, Arrays und Objekte

## Funktionen

1. **Einfacher Funktionsaufruf:**

Schreibe eine Funktion `greet`, die den Namen als Parameter annimmt und `"Hello, [name]!"` in die Konsole schreibt. Teste die Funktion mit verschiedenen Namen.

2. **Funktion mit Rückgabewert:**

Schreibe eine Funktion `add`, die zwei Zahlen als Parameter nimmt und deren Summe zurückgibt. Logge das Ergebnis.

3. **Funktion mit Standardwerten:**

Schreibe eine Funktion `multiply`, die zwei Zahlen multipliziert. Wenn der zweite Parameter nicht angegeben ist, soll der Standardwert 1 verwendet werden.

4. **Funktion mit Array als Argument:**

Schreibe eine Funktion `sumArray`, die ein Array von Zahlen als Parameter nimmt und die Summe der Zahlen zurückgibt.

```
const numbers = [2, 4, 6, 8];
```

5. **Funktion, die Objekte als Argumente verwendet:**

Schreibe eine Funktion `getFullName`, die ein Objekt mit `firstName` und `lastName` als Parameter nimmt und den vollständigen Namen zurückgibt.

```
const person = {
  firstName: "Jane",
  lastName: "Doe"
};
```

## Schleifen

6. **for-Schleife über ein Array:**

Schreibe eine Funktion `printArray`, die jedes Element eines Arrays mit einer for-Schleife in der Konsole ausgibt.

```
const colors = ["red", "green", "blue"];
```

7. **while-Schleife mit Abbruchbedingung:**

Schreibe eine Funktion `countDown`, die von einer gegebenen Zahl bis null herunterzählt und dabei jedes Mal die Zahl in der Konsole ausgibt.

8. **do-while-Schleife:**

Schreibe eine Funktion `askForPassword`, die in einer do-while-Schleife wiederholt das Passwort abfragt, bis das richtige Passwort eingegeben wird.

9. **Verschachtelte for-Schleifen:**

Schreibe eine Funktion `printMultiplicationTable`, die die Multiplikationstabelle von 1 bis 10 mit verschachtelten for-Schleifen in der Konsole ausgibt.

10. **for-Schleife mit Bedingung:**

Schreibe eine Funktion `printEvenNumbers`, die nur die geraden Zahlen in einem gegebenen Array mit einer for-Schleife ausgibt.

```
const nums = [1, 2, 3, 4, 5, 6];
```

## Arrays und Schleifen

11. **Summe von Array-Elementen:**

Schreibe eine Funktion `sumNumbers`, die ein Array von Zahlen als Parameter annimmt und die Summe der Zahlen berechnet. Verwende eine for-Schleife.

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

12. **Durchschnitt von Array-Elementen:**

Schreibe eine Funktion `averageNumbers`, die ein Array von Zahlen als Parameter annimmt und den Durchschnitt berechnet.

```
const numbers = [10, 20, 30, 40, 50];
```

13. **Maximalwert eines Arrays finden:**

Schreibe eine Funktion `findMax`, die das größte Element in einem Array von Zahlen findet.

```
const numbers = [10, 5, 8, 12, 3];
```

14. **Zählen von Vorkommen in einem Array:**

Schreibe eine Funktion `countOccurrences`, welches zwei Parameter nimmt, den Array und den Wert der gezählt werden soll und anschließend zählt, wie oft dieses Element in einem Array vorkommt.

```
const fruits = ["apple", "banana", "apple", "orange", "banana", "apple"];
```

15. **Array umkehren:**

Schreibe eine Funktion `reverseArray`, die die Elemente eines Arrays in umgekehrter Reihenfolge zurückgibt.

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

## Funktionen und Arrays kombinieren

16. **Funktion zur Manipulation von Array-Elementen:**

Schreibe eine Funktion `doubleNumbers`, die jedes Element eines Arrays verdoppelt und das neue Array zurückgibt.

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

17. **Funktion, die ein Array filtert:**

Schreibe eine Funktion `filterEvenNumbers`, die nur die geraden Zahlen aus einem gegebenen Array zurückgibt.

```
const numbers = [1, 2, 3, 4, 5, 6];
```

18. **Funktion, die Array-Elemente summiert, wenn sie eine Bedingung erfüllen:**

Schreibe eine Funktion `sumOddNumbers`, die die Summe aller ungeraden Zahlen in einem Array berechnet.

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

19. **Funktion, die prüft, ob ein Wert in einem Array enthalten ist:**

Schreibe eine Funktion `containsValue`, die prüft mit `.contains(value)`, ob ein bestimmter Wert in einem Array enthalten ist.

```
const fruits = ["apple", "banana", "orange"];
```

## Objekte und Arrays

20. **Objekt aus Array von Werten erstellen:**

Schreibe eine Funktion `createObjectFromArray`, die ein Array von Werten in ein Objekt umwandelt, wobei die Array-Indizes als Schlüssel verwendet werden.

```
const values = ["apple", "banana", "orange"];
```

21. **Funktion, die Eigenschaften eines Objekts in ein Array umwandelt:**

Schreibe eine Funktion `objectKeysToArray`, die die Schlüssel eines Objekts in ein Array umwandelt.

```
const person = {name: "John", age: 30, city: "Berlin"};
```

22. **Funktion, die alle Werte eines Objekts summiert:**

Schreibe eine Funktion `sumObjectValues`, die die Werte aller numerischen Eigenschaften eines Objekts summiert.

```
const data = {a: 10, b: 20, c: "hello"};
```

23. **Objekt aus zwei Arrays erstellen:**

Schreibe eine Funktion `createObjectFromTwoArrays`, die zwei Arrays (eines mit Schlüsseln und eines mit Werten) in ein Objekt umwandelt.

```
const keys = ["name", "age", "city"];
const values = ["John", 30, "Berlin"];
// Ausgabe: {name: "John", age: 30, city: "Berlin"}
```

24. **Objekte innerhalb eines Arrays summieren:**

Schreibe eine Funktion `sumPropertyInObjects`, die ein Array von Objekten nimmt und die Summe einer bestimmten Eigenschaft berechnet (z.B. `price` in einem Array von Produkten).

```
const products = [{name: "Product 1", price: 10}, {name: "Product 2", price: 20}, {name: "Product 3", price: 30}];
```

## Schleifen und Bedingungen kombinieren

25. **Funktion mit einer for-Schleife und if-Statement:**

Schreibe eine Funktion `countEvensAndOdds`, die die Anzahl der geraden und ungeraden Zahlen in einem Array zählt und das Ergebnis in einem Objekt zurückgibt.

```
const numbers = [1, 2, 3, 4, 5, 6];
```

26. **Verschachtelte Schleifen und Bedingungen:**

Schreibe eine Funktion `findDuplicates`, die doppelte Elemente in einem Array findet und sie in einem neuen Array speichert.

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

**27. Schleifen mit Array von Objekten:**

Schreibe eine Funktion `printAllProducts`, die alle Produkte in einem Array von Produktobjekten in der Konsole anzeigt. Jedes Produkt hat `name` und `price` als Eigenschaften.

```
const products = [{name: "Product 1", price: 10}, {name: "Product 2", price: 20}, {name: "Product 3", price: 30}];
```

## Fortgeschrittenere Aufgaben (mittleres Niveau)

**28. Funktion, die ein 2D-Array verarbeitet:**

Schreibe eine Funktion `sum2DArray`, die ein zweidimensionales Array als Parameter nimmt und die Summe aller Elemente berechnet.

```
const array2D = [[1, 2, 3], [4, 5, 6], [7, 8, 9]];
```

**29. Verschachtelte Schleifen für Paare von Array-Elementen:**

Schreibe eine Funktion `findAllPairs`, die alle Paare von Elementen in einem Array (z.B. [1, 2], [1, 3], [2, 3] usw.) in der Konsole anzeigt. Verwende verschachtelte for-Schleifen.

```
const numbers = [1, 2, 3, 4];  
// Ausgabe:  
// [1, 2]  
// [1, 3]  
// [1, 4]  
// [2, 3]
```

**30. Funktion, die Objekte auf Basis einer Bedingung filtert:**

Schreibe eine Funktion `filterProductsByPrice`, die ein Array von Produktobjekten und einen Maximalpreis als Parameter nimmt. Die Funktion gibt nur die Produkte zurück, deren Preis unter dem Maximalpreis liegt.

```
const products = [  
  {name: "Product 1", price: 50},  
  {name: "Product 2", price: 100},  
  {name: "Product 3", price: 150},  
  {name: "Product 4", price: 200}  
];  
  
console.log(filterProductsByPrice(products, 150));  
// Ausgabe:  
// [  
//   {name: "Product 1", price: 50},  
//   {name: "Product 2", price: 100}  
// ]
```