

# JavaScript Aufgaben zu Type Casting, Strings, Arrays, Objekten und Control Flow

---

## 1. Type Casting: String zu Number

Erstelle eine Variable `strNum` mit dem Wert `"123"`. Wandle diesen String in eine Zahl um und addiere 10 dazu. Logge das Ergebnis in die Konsole.

## 2. Type Casting: Number zu String

Erstelle eine Variable `num` mit dem Wert `456`. Wandle diese Zahl in einen String um und logge die Länge des resultierenden Strings.

## 3. String-Methode:

Erstelle eine Variable `greeting` mit dem Wert `"hello world"`. Wandle den String in Großbuchstaben um, und logge das Ergebnis.

## 4. String-Methode:

Erstelle eine Variable `phrase` mit dem Wert `"JavaScript is fun!"`. Schneide den Teilstring `JavaScript` heraus und in der Konsole anzuzeigen.

## 5. String-Methode:

Erstelle eine Variable `sentence` mit dem Wert `"This is a test"`. Verwende eine Methode, um den Satz in ein Array von Wörtern zu zerlegen, und logge das Array in die Konsole.

## 6. Array: Hinzufügen von Elementen

Erstelle ein Array `fruits` mit den Werten `["apple", "banana"]`. Füge das Wort `"orange"` zum Ende hinzu und `"kiwi"` an den Anfang. Logge das Array.

## 7. Array: Entfernen von Elementen

Erstelle ein Array `colors` mit den Werten `["red", "green", "blue"]`. Entferne das letzte Element und logge das Array.

## 8. Array: Zugriff auf Elemente

Erstelle ein Array `numbers` mit den Werten `[10, 20, 30, 40, 50]`. Greife auf das dritte Element zu und logge es.

Erstelle ein zweites Array mit den Werten von 60 bis 100 und vereine anschließend beide Arrays zu einem gemeinsamen Array. Drehe daraufhin die Anordnung der Zahlen um.

## 9. Objekte: Dot-Notation

Erstelle ein Objekt `car` mit den Eigenschaften `brand`, `model` und `year`. Greife mit der **Dot-Notation** auf die Eigenschaft `brand` zu und logge den Wert.

## 10. Objekte: Bracket-Notation

Verwende das Objekt `car` und greife mit der **Bracket-Notation** auf die Eigenschaft `model` zu. Logge den Wert.

Generiere für das `car` Objekt anschließend die Eigenschaft `leftSteering` und gib ihr einen Boolean Wert Deiner Wahl. Schau Dir das Objekt danach in der Konsole an.

## 11. Objekte: Eigenschaften hinzufügen (Dot-Notation)

Verwende die **Dot-Notation**, um dem Objekt `car` eine neue Eigenschaft `color` mit dem Wert `"red"` hinzuzufügen. Logge das aktualisierte Objekt.

## 12. Objekte: Eigenschaften hinzufügen (Bracket-Notation)

Verwende die **Bracket-Notation**, um dem Objekt `car` eine neue Eigenschaft `price` mit dem Wert `20000` hinzuzufügen. Logge das aktualisierte Objekt.

## 13. if-Statement

Erstelle eine Funktion `checkEven`, die eine Zahl als Parameter annimmt. Wenn die Zahl gerade ist, soll `"Die Zahl ist gerade"` ausgegeben werden, andernfalls `"Die Zahl ist ungerade"`. Teste die Funktion mit der Zahl `4`.

## 14. if-else if-else-Statement

Erstelle eine Funktion `gradeClassifier`, die eine Zahl als Parameter annimmt (1 bis 100). Klassifiziere die Zahl:

- 90 und höher: `"A"`

- 80 bis 89: "B"
- 70 bis 79: "C"
- 60 bis 69: "D"
- Unter 60: "F" Teste die Funktion mit der Zahl 85 .

## 15. switch-Statement

Erstelle eine Funktion `getDayOfWeek` , die eine Zahl (1 bis 7) als Parameter annimmt und den Wochentag zurückgibt. Verwende ein `switch` -Statement, um die Tage zuzuordnen (1 = Montag, 2 = Dienstag, usw.). Teste die Funktion mit der Zahl 3 .

## 16. Objekte: Funktion im Objekt definieren

Erstelle ein Objekt `student` mit den Eigenschaften `firstName` , `lastName` und einer Methode `getFullName` , die den vollständigen Namen zurückgibt. Logge das Ergebnis der Methode.

## 17. Objekte: Alle Keys abfragen

Erstelle ein Objekt `book` mit den Eigenschaften `title` , `author` , `pages` und `published` . Benutze eine Methode, um alle Schlüssel (keys) des Objekts zu erhalten und logge das Ergebnis.

## 18. Objekte: Alle Values abfragen

Verwende das Objekt `book` und verwende eine Methode um alle Werte des Objekts zu erhalten. Logge das Ergebnis.

## 19. Objekte: Entries abfragen

Verwende das Objekt `book` und verwende eine Methode um alle Schlüssel-Werte-Paare als Arrays zu erhalten. Logge das Ergebnis.

## 20. Objekte: Dynamische Schlüssel in Bracket-Notation

Erstelle eine Funktion `getProperty` , die zwei Parameter annimmt: ein Objekt und den Namen einer Eigenschaft (als String). Verwende die **Bracket-Notation**, um den Wert der übergebenen Eigenschaft zu erhalten und zu loggen.