

# LU01c - Arrays und Objekte

In JavaScript sind Arrays und Objekte die beiden wichtigsten Datentypen für die Speicherung und Strukturierung von Daten. Während Arrays geordnete Sammlungen von Elementen sind, bieten Objekte eine Möglichkeit, Daten in Schlüssel-Wert-Paaren zu organisieren. Beide Datentypen sind äußerst flexibel und ermöglichen komplexe Datenstrukturen und Operationen.

## Arrays

Ein Array ist eine geordnete Liste von Werten, wobei jedes Element über einen numerischen Index angesprochen wird. Arrays können jede Art von Daten enthalten, einschließlich anderer Arrays und Objekte.

### Erstellen eines Arrays

Ein Array kann auf verschiedene Arten erstellt werden:

```
const zahlen = [1, 2, 3, 4, 5];
const gemischt = [1, "Hallo", true, null, undefined];
```

In diesen Beispielen wird das Array `zahlen` mit einer Liste von Zahlen und das Array `gemischt` mit Werten unterschiedlicher Datentypen erstellt.

### Zugriff auf Array-Elemente

Um auf ein bestimmtes Element in einem Array zuzugreifen, verwendest du den Index des Elements, wobei der Index bei 0 beginnt:

```
console.log(zahlen[0]); // 1
console.log(gemischt[1]); // "Hallo"
```

### Methoden zum Hinzufügen neuer Werte in ein Array

#### **push(): Am Ende des Arrays hinzufügen**

Die Methode **push()** fügt ein oder mehrere Elemente an das Ende des Arrays an.

```
zahlen.push(6);
console.log(zahlen); // [1, 2, 3, 4, 5, 6]
```

## unshift(): Am Anfang des Arrays hinzufügen

Die Methode **unshift()** fügt ein oder mehrere Elemente an den Anfang des Arrays an.

```
zahlen.unshift(0);
console.log(zahlen); // [0, 1, 2, 3, 4, 5, 6]
```

## Direkte Zuweisung eines Werts

Du kannst auch direkt einen Wert an einer bestimmten Position eines Arrays zuweisen, indem du den Index angibst.

```
zahlen[7] = 7;
console.log(zahlen); // [0, 1, 2, 3, 4, 5, 6, 7]
```

## splice(): An beliebiger Position hinzufügen oder löschen

Die Methode `splice()` erlaubt das Einfügen, Ersetzen oder Löschen von Elementen an einer beliebigen Stelle im Array.

```
zahlen.splice(3, 0, 10); // An Index 3 einfügen, ohne Elemente zu löschen
console.log(zahlen); // [0, 1, 2, 10, 3, 4, 5, 6, 7]
```

## concat(): Arrays kombinieren

Die Methode `concat()` kombiniert zwei oder mehr Arrays zu einem neuen Array.

```
const mehrZahlen = [8, 9];
const alleZahlen = zahlen.concat(mehrZahlen);
console.log(alleZahlen); // [0, 1, 2, 10, 3, 4, 5, 6, 7, 8, 9]
```

## Spread-Syntax (...): Ein Array in ein anderes Array einfügen

Die Spread-Syntax ermöglicht das Einfügen eines Arrays in ein anderes Array:

```
const nochMehrZahlen = [11, 12];
const neuesArray = [...alleZahlen, ...nochMehrZahlen];
console.log(neuesArray); // [0, 1, 2, 10, 3, 4, 5, 6, 7, 8, 9, 11, 12]
```

# Objekte

Ein Objekt in JavaScript ist eine Sammlung von Schlüssel-Wert-Paaren. Jeder Schlüssel (oder auch

„Property“) ist ein eindeutiger Bezeichner, und jeder Wert kann ein beliebiger Datentyp sein, einschliesslich anderer Objekte oder Arrays.

## Erstellen eines Objekts

Ein Objekt kann durch eine einfache Notation mit geschweiften Klammern erstellt werden:

```
const person = {  
    name: "Max",  
    alter: 25,  
    beruf: "Entwickler"  
};
```

In diesem Beispiel hat das Objekt person drei Eigenschaften: name, alter und beruf, denen entsprechende Werte zugewiesen sind.

## Zugriff auf Eigenschaften

Du kannst auf die Eigenschaften eines Objekts entweder über den Punkt-Operator oder die eckigen Klammern zugreifen:

```
console.log(person.name); // "Max"  
console.log(person["alter"]); // 25
```

## Methoden zum Hinzufügen oder Ändern von Werten in einem Objekt

### Direkte Zuweisung

Du kannst einem Objekt neue Eigenschaften hinzufügen oder bestehende Eigenschaften ändern, indem du einfach einen Wert zuweist.

```
person.nationalitaet = "Schweizer";  
console.log(person.nationalitaet); // "Schweizer"
```

### Spread-Syntax für Objekte

Ähnlich wie bei Arrays kannst du mit der Spread-Syntax ein Objekt duplizieren und neue Eigenschaften hinzufügen.

```
const neuePerson = { ...person, geschlecht: "männlich" };  
console.log(neuePerson);  
// { name: 'Max', alter: 25, beruf: 'Entwickler', nationalitaet:  
// 'Schweizer', geschlecht: 'männlich' }
```

## Object.assign(): Objekte kombinieren

Die Methode Object.assign() kopiert die Eigenschaften eines oder mehrerer Objekte in ein Zielobjekt.

```
const weitereEigenschaften = { hobby: "Schwimmen", sprache: "Deutsch" };
Object.assign(person, weitereEigenschaften);
console.log(person);
// { name: 'Max', alter: 25, beruf: 'Entwickler', nationalitaet: 'Schweizer', hobby: 'Schwimmen', sprache: 'Deutsch' }
```

From:  
<https://wiki.bzz.ch/> - BZZ - Modulwiki



Permanent link:  
<https://wiki.bzz.ch/modul/ffit/js/learningunits/lu01/arrays-und-objekte>

Last update: **2024/10/24 07:26**