

JS Objekte und Arrays

Lernziele

- Erstellen, Zugreifen auf und Manipulieren von Arrays
 - Erstellen, Zugreifen auf und Manipulieren von Objekten
 - Wissen, wie man Eigenschaften und Methoden von Objekten durch Logging findet
-

Arrays

Arrays sind ein strukturierter Datentyp, der mehrere Werte in einer Variable speichern kann.

Du kannst ein Array mit `[]` eckigen Klammern (Array-Literale) deklarieren:

```
const einkaufsliste = ["Apfel", "Tomate"];
```

Jedes Element im array hat einen Index, der bei 0 beginnt. Du kannst einzelne Elemente mit der Index-Notation und dem Index des Elements zugreifen:

```
einkaufsliste[0]; // "Apfel"  
einkaufsliste[1]; // "Tomate"
```

Arrays können jeden Typ von Werten enthalten, sogar ein anderes array. Dies wird als verschachteltes Array bezeichnet. Die Werte von verschachtelten Arrays können abgerufen werden, indem zuerst der Index des verschachtelten Arrays ausgewählt und dann der Index des Elements im verschachtelten Array angegeben wird.

```
const verschachteltesArray = ["a", 1, ["a", "neues", "Satz"], false];  
verschachteltesArray[2][1]; // "neues"
```

Du kannst einzelne Werte in einem Array überschreiben:

```
const einkaufsliste = ["Apfel", "Tomate"];  
einkaufsliste[0] = "Banane";  
einkaufsliste; // ["Banane", "Tomate"];
```

Häufige Array-Attribute und -Methoden

Attribute / Method	Effect
--------------------	--------

Attribute / Method	Effect
<code>array.length</code>	gibt die Anzahl der Elemente im Array zurück
<code>array.push(element)</code>	fügt <code>element</code> am Ende des Arrays hinzu
<code>array.pop()</code>	entfernt das letzte Element eines Arrays
<code>array.unshift(element)</code>	fügt <code>element</code> als erstes Element des Arrays hinzu
<code>array.shift()</code>	entfernt das erste Element des Arrays

💡 Es gibt noch viele weitere Array-Methoden und -Attribute, die wir in späteren Sitzungen entdecken werden. Siehe die [MDN Docs](#) für weitere Informationen.

Objekte

Objekte sind ein strukturierter Datentyp, der ihre Werte nicht an einen Index, sondern an einen einzigartigen Schlüssel koppelt.

Du kannst ein Objekt mit `{}` geschweiften Klammern (Objekt-Literale) deklarieren:

```
const person = {  
  name: "Max Paddington",  
  alter: 21,  
  istStudent: false,  
};
```

Du kannst auf die Eigenschaften mit der Punkt-Notation zugreifen:

```
person.name; //"Max Paddington"
```

Du kannst auch auf die Eigenschaften mit der Index-Notation zugreifen:

```
person["alter"]; // 21
```

Objekte können verschachtelt sein:

```
const person = {  
  name: "Max Paddington",  
  alter: 21,  
  istStudent: false,  
  adresse: {  
    straße: "Berliner Str.",  
    hausnummer: 42,  
    stadt: "Leipzig",  
  },  
};
```

```
    plz: "12345",  
  },  
};
```

Verschachtelte Werte können durch Verkettung der Punkt-Notation und/oder der Index-Notation abgerufen werden.

```
person.adresse.straße; // "Berliner Str."  
person.adresse["stadt"]; // "Leipzig"
```

Du kannst Werte von Objekt-Eigenschaften ändern, indem du sie mit der Punkt- oder Index-Notation neu zuweist:

```
person.name = "Max Paddington";  
person["alter"] = 33;
```

Du kannst auf dieselbe Weise neue Eigenschaften hinzufügen:

```
person.punktzahl = 15;
```

Du kannst Eigenschaften mit dem delete-Schlüsselwort löschen:

```
delete person.punktzahl;
```

Verschachtelte Objekte / Arrays

Arrays können Objekte enthalten und umgekehrt:

```
const personenArray = [  
  {  
    name: "John",  
    alter: 22,  
  },  
  {  
    name: "Alex",  
    alter: 33,  
  },  
];
```

```
const benutzer = {  
  benutzerId: "1234",  
  mail: "test@mail.com",  
  einkaufswagen: ["Tomate", "Banane", "Schokolade"],  
};
```

Du kannst Elemente über verkettete Punkt- / Index-Notation abrufen:

```
personenArray[1].name; // "Alex"  
benutzer.einkaufswagen[0]; // "Tomate"
```

Ressourcen

Array

[MDN Docs: Array](#)

Object

[MDN Docs: Object](#)