

Arrays in JavaScript

Typeof gibt einfach object aus,
instanceof Array gibt ein true
zurück

students.push verändert das
Array students

students.pop entfernt nicht nur
das letzte Element und definiert
das Array students neu,

```
/**
 * Mit einem Array können wir z.B. eine Teilnehmerl
iste
 * speichern!
 */
let students = [
  "Max Müller",
  "Erika Mustermann",
  "Annika Müller",
  "Max Mustermann"
]
console.log(students)

// Was ist der Typ von einem Array?
console.log(typeof students)
console.log(students instanceof Array)

// Wie viele Elemente sind in dem Array?
console.log(".length", students.length)

// Auf ein einzelnes Element zugreifen
console.log("[0]", students[0])

// Wenn ein Element nicht existiert, wird dies
// ignoriert, und es wird nur "undefined" zurückgeg
eben
console.log("[100]", students[100])

// Element hinzufügen
students.push("Joker", "Joker 2")
console.log(students)

// Letztes Element entfernen
const lastElement = students.pop()
console.log("lastElement", lastElement)
console.log(students)

// Ein Element überschreiben
students[0] = "Maximilian Müller"
console.log(students)
```

sondern gibt auch das gerade
entfernte Element zurück.

Arrays ordnen, Elemente hinzufügen, wegnehmen

```
"use strict"
```

```
let students = [  
  "Max Müller",  
  "Erika Mustermann",  
  "Annika Müller",  
  "Max Mustermann"  
]
```

```
// Prüfen: Ist ein Element im Array vorhanden?  
console.log(students.indexOf("Max Müller"))
```

```
// Wenn ein Element nicht existiert -> -1!  
console.log(students.indexOf("Max Müller!"))
```

```
// Arrays sortieren  
students.sort()  
console.log(students)
```

```
// Sortierung umdrehen  
students.reverse()  
console.log(students)
```

```
// Wichtig: Sparse Arrays!  
//students[30] = "Joker"  
console.log(students)
```

```
// Die .splice()-Funktion zum Entfernen von Elementen (Anfang 0, 2 Elemente)  
students.splice(0, 2)  
console.log(students)
```

```
// Die .splice()-Funktion zum Hinzufügen von Elementen  
students.splice(1, 0, "Joker")  
console.log(students)
```

Aufgabe Arrays

Ein Array mit einer Teilnehmerliste ist schon vorhanden und soll in JavaScript um weitere Teilnehmer ergänzt werden können.

Teilnehmerliste

1. Koch, Franz

2. Meyer, Lutz

3. Müller, Manuel

4. Renz, Carol

Teilnehmer hinzufügen:

Nachname, Vorname

☐ AGB gelesen und akzeptiert?

Teilnehmer hinzufügen!

Die neu eingetragenen Teilnehmer sollen in einem Array (kurzfristig) gespeichert werden.

Die Liste soll alphabetisch sortiert werden und der eingetragene Teilnehmer an der richtigen Stelle hinzukommen.

Schritt 1 – Array anlegen

```
let students = ["Müller, Manuel", "Koch, Franz", "Meyer, Lutz"];
```

Schritt 2 – einen weiteren Teilnehmer hinzufügen

```
students.push('Renz, Carol');
```

Schritt 3 – Variablen anlegen

```
const addStudentButton = document.getElementById("add-student")
const nameInput = ???
const studentList = ???
const agbInput = ???
```

Schritt 4 – Die eingegebenen Elemente des Array in einer Liste innerhalb studentList ausgeben.

Verwenden Sie sinnvoll z. B. diese Bausteine

```
function createListItem(x) {
    let nameNeu = document.createTextNode(x);

    // ein leeres Listenelement erzeugen

    liElement.appendChild(nameNeu)
    liElement.classList.add("list-group-item")

    //das Listenelement der Liste studentList hinzufügen (appendChild)

}

function createList() {

    //Liste sortieren
    //for Schleife über alle Elemente und für diese einen Listeneintrag erzeugen

}
```

Schritt 5 – die neuen Personen aufnehmen in die Liste und abfragen, ob Eingabe vorhanden ist und die Checkbox für die AGB's angeklickt wurde

```
addStudentButton.addEventListener("click", () => {

    if (agbInput.checked === false) {
        alert("Bitte AGB lesen!")
        return
    }
    let text = nameInput.value

    if (text === "") {
        agbInput.checked = false;
        return
    }

    //neues Array erzeugen mit dem eingetragenen neuen Namen
    //Liste am Bildschirm neu erzeugen (Funktion?)
    //Werte zurücksetzen

})
```