JS Objects and Arrays

Learning Objectives

- Creating, accessing, and manipulating arrays
- Creating, accessing, and manipulating objects
- Knowing how to find properties and methods of objects by logging

---

Arrays

Arrays are a structured data type which can store multiple values in one variable.

You can declare an array using `[]` square brackets (array literals):

```
const shoppingList = ["apple", "tomato"];
```

Each item in the array has an index, which starts at 0. You can access individual items using the bracket notation and the item's index:

```
shoppingList[0]; // "apple"
shoppingList[1]; // "tomato"
```

Arrays can hold any type of value, even another array. This is called a nested array. The values of nested arrays can be accessed by choosing the index of the nested array first and then stating the index of the element inside the nested array.

```
const nestedArray = ["a", 1, ["a", "new", "sentence"], false];
nestedArray[2][1]; // "new"
```

You can overwrite individual values in an array:

```
const shoppingList = ["apple", "tomato"];
shoppingList[0] = "banana";
shoppingList; // ["banana","tomato"];
```

Common Array Attributes and Methods

Attribute / Method	Effect
`array.length`	returns the number of elements in the array

Attribute / Method	Effect
`array.push(element)`	adds `element` to the end of the array
`array.pop()`	removes the last element of an array
`array.unshift(element)`	adds `element` as the first element of the array
`array.shift()`	removes the first element of the array

> 💡 There are much more array methods and attributes which we will discover in later sessions. Go to the MDN Docs for more information.

Objects

Objects are a structured data type, which couple their values not to an index, but to a unique key.

You can declare an object using `{}` curly brackets (object literals):

```
const person = {
  name: "Max Paddington",
  age: 21,
  isStudent: false,
};
```

You can access the properties using the dot notation:

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

You can also access the properties using the bracket notation:

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

Objects can be nested:

```
const person = {
  name: "Max Paddington",
  age: 21,
  isStudent: false,
  address: {
    street: "Berliner Str.",
    houseNumber: 42,
    city: "Leipzig",
    zipCode: "12345",
```

```
  },
};
```

Nested values can be accessed by chaining the dot notation and/or the bracket notation together.

```
person.address.street; // "Berliner Str."
person.address["city"]; // "Leipzig"
```

You can change values of object properties by reassigning them using the dot or bracket notation:

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

You can add new properties in the same way:

```
person.score = 15;
```

You can delete properties using the delete keyword:

```
delete person.score;
```

Nested Objects / Arrays

Arrays can contain objects and vice versa:

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

```
const user = {
  userId: "1234",
  mail: "test@mail.com",
```

```
  shoppingCart: ["tomato", "banana", "chocolate"],
};
```

You can access elements via chained dot / bracket notation:

```
peopleArray[1].name; // "Alex"
user.shoppingCart[0]; // "tomato"
```

Resources

Array

MDN Docs: Array

Object

MDN Docs: Object