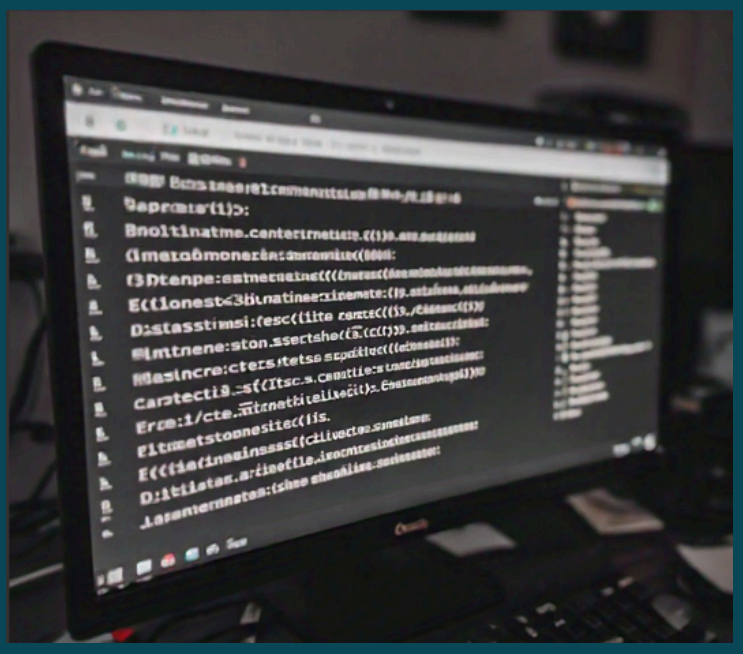
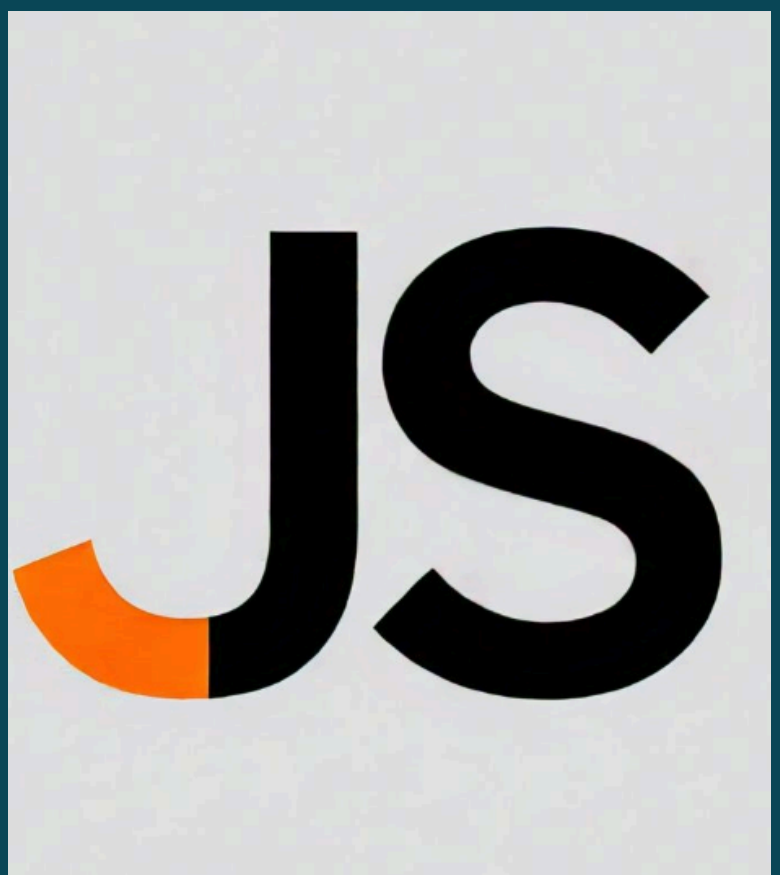
JAVASCRIPT



ARRAY METHODS



Array.of()

Creates an array from given values.

```
const faces = Array.of("😊", "🤔", "😎");
// ["😊", "🤔", "😎"]
```

Array.from()

Converts an iterable into an array.

```
const faces = Array.from("😄😇😅");
// ["😄", "😇", "😅"]
```

fill()

Fills the array with a specific value.

```
const faces = new Array(3).fill("😮");
// ["😮", "😮", "😮"]
```

push()

Adds elements to the end of the array.

```
const faces = ["😊", "🤔"];
faces.push("😎"); // ["😊", "🤔", "😎"]
```

pop()

Removes the last element.

```
const faces = ["😊", "🤔", "😎"];
faces.pop(); // ["😊", "🤔"]
```

unshift()

Adds elements to the beginning.

```
const faces = ["🤔", "😎"];
faces.unshift("😊"); // ["😊", "🤔", "😎"]
```

shift()

Removes the first element.

```
const faces = ["😊", "🤔", "😎"];
faces.shift(); // ["🤔", "😎"]
```

indexOf()

Finds the index of an element.

```
const faces = ["😊", "🤔", "😎"];
const index = faces.indexOf("🤔"); // 1
```

includes()

Checks if an array contains an element.

```
const faces = ["😊", "🤔", "😎"];
const hasFace = faces.includes("😎"); // true
```

find()

Finds the first element matching a condition.

```
const faces = ["😊", "🤔", "😎"];
const face = faces.find(face => face === "🤔");
// "🤔"
```

findIndex()

Finds the index of the first element matching a condition.

```
const faces = ["😊", "🤔", "😎"];
const index = faces.findIndex(face => face === "😎");
// 2
```

slice()

Returns a shallow copy of a portion of the array.

```
const faces = ["😊", "🤔", "😎"];
const sliced = faces.slice(1, 3); // ["🤔", "😎"]
```

splice()

Adds or removes elements in the array.

```
const faces = ["😊", "🤔", "😎"];
faces.splice(1, 1, "😇"); // ["😊", "😇", "😎"]
```

concat()

Combines two or more arrays.

```
const faces1 = ["😊"];
const faces2 = ["🤔", "😎"];
const allFaces = faces1.concat(faces2);
// ["😊", "🤔", "😎"]
```

join()

Joins array elements into a string.

```
const faces = ["😊", "🤔", "😎"];
const result = faces.join(" - ");
// "😊 - 🤔 - 😎"
```

sort()

Sorts the elements of an array.

```
const faces = ["😎", "😊", "🤔"];
faces.sort();
// ["😊", "🤔", "😎"]
```

reverse()

Reverses the array.

```
const faces = ["😊", "🤔", "😎"];
faces.reverse();
// ["😎", "🤔", "😊"]
```

map()

Transforms each element using a callback.

```
const faces = ["😊", "🤔", "😎"];
const transformed = faces.map(face => face + "🔥");
// ["😊🔥", "🤔🔥", "😎🔥"]
```

forEach()

Executes a callback for each element.

```
const faces = ["😊", "🤔", "😎"];
faces.forEach(face => console.log(face));
// Logs each emoji
```

filter()

Transforms each element using a callback.

```
const faces = ["😊", "🤔", "😎"];
const filtered = faces.filter(face => face !== "🤔");
// ["😊", "😎"]
```

reduce()

Reduces the array to a single value.

```
const faces = ["😊", "🤔", "😎"];
const combined = faces.reduce((acc, face) => acc + face, "");
// "😊🤔😎"
```

every()

Checks if all elements match a condition.

```
const faces = ["😊", "🤔", "😎"];
const allAreStrings = faces.every(
                        face => typeof face === "string"
                      );
// true
```