Study sheet   
  
**JavaScript Study Guide**

**1. Array Methods**

**map()**

* Returns a **new array** with each element transformed by the callback function.
* Syntax: array.map((element, index, array) => { /\* transformation \*/ })
* Example:

const nums = [1, 2, 3];

const doubled = nums.map(n => n \* 2); // [2, 4, 6]

**filter()**

* Returns a **new array** with only elements that pass a condition.
* Syntax: array.filter((element, index, array) => condition)
* Example:  
  const nums = [1, 2, 3, 4];

const evens = nums.filter(n => n % 2 === 0); // [2, 4]

**forEach()**

* Iterates over an array but **does not return** anything.
* Used for performing actions like logging or modifying elements.
* Example:
* const nums = [1, 2, 3];

nums.forEach(n => console.log(n));

**reduce()**

* Reduces an array to a single value.
* Syntax: array.reduce((accumulator, element, index, array) => { /\* operation \*/ }, initialValue)
* Example:
* const nums = [1, 2, 3, 4];

const sum = nums.reduce((total, n) => total + n, 0); // 10

**2. Working with Objects**

**Object Destructuring**

* Extract properties from an object into variables.
* Example:
* const person = { name: "Alice", age: 25 };
* const { name, age } = person;

console.log(name, age); // Alice 25

**Rest Operator with Objects**

* Collects remaining properties into a separate object.
* Example:
* const person = { name: "Alice", age: 25, city: "New York" };
* const { name, ...rest } = person;

console.log(rest); // { age: 25, city: "New York" }

**Spread Operator with Objects**

* Copies and merges objects.
* Example:
* const obj1 = { a: 1, b: 2 };
* const obj2 = { b: 3, c: 4 };
* const merged = { ...obj1, ...obj2 };

console.log(merged); // { a: 1, b: 3, c: 4 }

**3. Working with Arrays**

**Array Destructuring**

* Extract values from an array into variables.
* Example:
* const nums = [1, 2, 3, 4];
* const [a, b, c, d] = nums;

console.log(b, d); // 2 4

**Rest Operator with Arrays**

* Collects remaining elements into a new array.
* Example:
* const nums = [1, 2, 3];
* const [first, ...rest] = nums;

console.log(rest); // [2, 3]

**Spread Operator with Arrays**

* Merges or copies arrays.
* Example:
* const arr1 = [1, 2, 3];
* const arr2 = [4, 5];
* const combined = [...arr1, ...arr2];

console.log(combined); // [1, 2, 3, 4, 5]

**4. DOM Manipulation**

**Selecting Elements**

* document.querySelector("#id") - Selects an element by ID.
* document.getElementById("id") - Selects an element by ID.

**Modifying Elements**

* Change HTML content:

document.getElementById("container").innerHTML = "<p>Hello</p>";

* Create and append elements:
* const div = document.createElement("div");
* div.textContent = "New Element";

document.body.appendChild(div);

**5. Promises & Async/Await**

**fetch() and Promises**

* fetch() returns a promise that resolves to a response object.
* Example:
* fetch("https://api.example.com/data")
* .then(response => response.json())

.then(data => console.log(data));

**Async/Await**

* Used to handle asynchronous code more easily.
* Example:
* async function getData() {
* let response = await fetch("https://api.example.com/data");
* let data = await response.json();
* console.log(data);

}

**6. Template Literals**

* Use backticks (`) to embed expressions.
* Example:
* const name = "Alice";

console.log(`Hello, ${name}!`); // "Hello, Alice!"

**7. Summary of Key Concepts**

|  |  |
| --- | --- |
| **Concept** | **Description** |
| .map() | Transforms an array into a new array |
| .filter() | Returns an array of elements that pass a condition |
| .forEach() | Loops through an array but returns undefined |
| .reduce() | Reduces an array to a single value |
| Object Destructuring | Extracts properties from an object |
| Spread Operator (...) | Copies or merges arrays/objects |
| Rest Operator (...rest) | Gathers remaining values into an array/object |
| fetch() | Returns a Promise for making API requests |
| async/await | Handles asynchronous operations cleanly |
| Template Literals | Use backticks to embed expressions |

Study these concepts and practice coding examples to reinforce your understanding!