Modern JavaScript Cheat Sheet

Table of Contents

---

1. Variables

```
let variableName = "Hello, JavaScript!";
const constantValue = 42;
```

2. Data Types

```
// String, Number, Boolean
const name = "Najem";
const age = 25;
const isStudent = true;

// Array, Object
const colors = ['red', 'green', 'blue'];
const person = { name: 'John', age: 30 };

// null, undefined
let value1 = null;
let value2;
```

3. Arrays

```
const colors = ['red', 'green', 'blue'];

// Access element
const firstColor = colors[0]; // 'red'

// Add an element
colors.push('yellow'); // ['red', 'green', 'blue', 'yellow']

// Loop through elements
for (const color of colors) {
  console.log(color);
}
```

4. Array Functions

```
// Map
const numbers = [1, 2, 3, 4, 5];
const doubled = numbers.map((num) => num * 2); // [2, 4, 6, 8, 10]

// Filter
const evenNumbers = numbers.filter((num) => num % 2 === 0); // [2, 4]

// Reduce
const sum = numbers.reduce((acc, num) => acc + num, 0); // 15

// forEach
colors.forEach((color) => console.log(color));
// Output:
// 'red'
// 'green'
// 'blue'

// find and findIndex
const users = [
  { id: 1, name: 'Alice' },
  { id: 2, name: 'Bob' },
  { id: 3, name: 'Charlie' },
];
const user = users.find((user) => user.id === 2); // { id: 2, name: 'Bob' }
const userIndex = users.findIndex((user) => user.id === 2); // 1
```

5. Objects

```
const person = {
  name: 'John',
  age: 30,
};
```

```
// Access property
const name = person.name; // 'John'

// Add a new property
person.city = 'New York';

// Loop through properties
for (const key in person) {
  console.log(key, person[key]);
}
// 'name' 'John'
// 'age' 30
// 'city' 'New York'
```

6. Functions

```
function greet(name) {
  return `Hello, ${name}!`;
}

// Function expression
const add = function (a, b) {
  return a + b;
};

// Arrow function
const multiply = (a, b) => a * b;
```

7. Arrow Functions

```
const square = (x) => x * x;
const double = (x) => {
  return x * 2;
};
```

8. Template Literals

```
const name = 'Alice';
const greeting = `Hello, ${name}!`; // 'Hello, Alice!'
```

9. Destructuring

```
const person = { name: 'Bob', age: 25 };
const { name, age } = person; // name = 'Bob', age = 25

const colors = ['red', 'green', 'blue'];
const [first, second] = colors; // first = 'red', second = 'green'
```

10. Loops

```
// For Loop
for (let i = 0; i < 5; i++) {
  console.log(i);
}
// For...of Loop
const colors = ['red', 'green', 'blue'];
for (const color of colors) {
  console.log(color);
}
```

11. Conditional Statements

```
const age = 18;

if (age >= 18) {
  console.log('You are an adult.'); // 'You are an adult.'
} else {
  console.log('You are a minor.');
}
```

12. Promises

```
const fetchData = () => {
  return new Promise((resolve, reject) => {
    // Async operation
    if (success) {
      resolve(data);
    } else {
      reject(error);
    }
  });
};
```

13. Async/Await

```
async function getData() {
  try {
    const response = await fetchData();
    console.log(response);
  } catch (error) {
    console.error(error);
  }
}
```

14. Fetch API

```
fetch('https://api.example.com/data')
  .then((response) => response.json())
  .then((data) => console.log(data))
  .catch((error) => console.error(error));
```

15. DOM Selection and Manipulation

Selecting Elements

```
const elementById = document.getElementById('elementId');
const elementsByClass = document.getElementsByClassName('className');
const elementsByTag = document.getElementsByTagName('tagName');
const elementBySelector = document.querySelector('CSSSelector');
const elementsBySelectorAll = document.querySelectorAll('CSSSelector');
```

Manipulating Elements

```
// Changing text content
elementById.textContent = 'New Text';

// Changing HTML content
elementById.innerHTML = '<strong>New HTML</strong>';

// Adding/removing CSS classes
elementById.classList.add('newClass');
elementById.classList.remove('oldClass');

// Creating new elements
const newElement = document.createElement('tagName');

// Appending elements
parentElement.appendChild(newElement);

// Removing elements
parentElement.removeChild(childElement);

// Event handling
elementById.addEventListener('click', (event) => {
  // Handle click event
});
```