

Online JAVASCRIPT CHEAT SHEET

JS




WEB DEVELOPMENT



Sekhar Metla
Author



LEARN NOW

-  Quick skills
-  learn from anywhere
-  www.harisystems.com

Future Skills
www.harisystems.com

JavaScript cheat sheet document for quick remember keywords and best practices few examples: JavaScript Cheat Sheet

1. Basics

- Variables (var, let, const)
- Data Types (Number, String, Boolean, Object, Array, null, undefined)
- Operators (+, -, *, /, %, ++, --, =, ==, ===, !=, !==, &&, ||, !, etc.)
- Conditionals (if, else if, else, switch)
- Loops (for, while, do-while, for...in, for...of)

2. Functions

- Declaring Functions
- Function Expressions
- Arrow Functions
- Parameters and Arguments
- Return Statement
- Higher-order Functions (map, filter, reduce, etc.)

3. Arrays

- Array Declaration
- Accessing Elements
- Adding and Removing Elements
- Array Methods (push, pop, shift, unshift, splice, slice, etc.)

4. Objects

- Object Literal
- Accessing Object Properties
- Adding and Modifying Properties
- Nested Objects
- Object Methods

5. DOM Manipulation

- Selecting Elements (getElementById, querySelector, etc.)
- Changing HTML Content (innerHTML,.textContent, etc.)
- Modifying Styles (style property)
- Adding and Removing Classes (classList)
- Handling Events (addEventListener)

6. Asynchronous JavaScript

- setTimeout and setInterval
- Promises
- async/await

7. Error Handling

- try...catch
- throw

8. Local Storage

- Saving and Retrieving Data

9. JSON (JavaScript Object Notation)

- JSON.stringify
- JSON.parse

10. Regular Expressions

- Creating Patterns
- Matching and Replacing Text

11. Scope and Closures

- Global Scope
- Function Scope
- Block Scope
- Closures

12. Classes and Object-Oriented Programming

- Class Declaration
- Constructor
- Methods
- Inheritance (extends, super)

13. Fetch API

- Making HTTP Requests

14. Modules (ES6)

- Exporting and Importing Modules

15. Browsers and Window Object

- Browser Information (navigator)
- Popup Dialogs (alert, confirm, prompt)

16. Debugging JavaScript

- Using console.log
- Browser Developer Tools

17. Best Practices

- Avoiding Global Variables
- Writing Readable Code
- Handling Asynchronous Code
- Performance Tips

JavaScript cheat sheet document with few examples for each concept:

JavaScript Cheat Sheet with Examples

1. Variables and Data Types

```
// Number  
let age = 30;
```

```
// String  
let name = "John";
```

```
// Boolean  
let isStudent = true;
```

```
// Array  
let fruits = ["apple", "banana", "orange"];
```

```
// Object  
let person = {  
  name: "Alice",  
  age: 25,  
  isStudent: false  
};
```

2. Basic Math Operations

```
let x = 10;  
let y = 5;  
  
let sum = x + y;  
let difference = x - y;  
let product = x * y;  
let quotient = x / y;  
let remainder = x % y;
```

3. Conditional Statements

```
let grade = 85;  
  
if (grade >= 90) {  
  console.log("A");  
} else if (grade >= 80) {  
  console.log("B");  
} else {  
  console.log("C");  
}
```

4. Loops

```
// For Loop  
for (let i = 1; i <= 5; i++) {  
  console.log(i);  
}
```

```
// While Loop  
let count = 1;  
while (count <= 5) {  
  console.log(count);  
  count++;  
}
```

5. Functions

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

```
let greeting = greet("Alice");  
console.log(greeting);
```

6. Arrays

```
let fruits = ["apple", "banana", "orange"];
```

```
// Accessing elements  
console.log(fruits[0]); // Output: "apple"
```

```
// Adding elements  
fruits.push("grape");
```

```
// Removing elements  
fruits.pop();
```

7. Objects

```
let person = {  
  name: "John",  
  age: 30,  
  isStudent: true  
};
```

```
// Accessing properties  
console.log(person.name); // Output: "John"
```

```
// Adding properties  
person.city = "New York";
```

```
// Modifying properties  
person.age = 31;
```

8. DOM Manipulation

```
// HTML: <p id="demo">This is a  
paragraph.</p>  
let paragraph =  
document.getElementById("demo");
```

```
paragraph.textContent = "Updated
paragraph";
```

9. Event Handling

```
// HTML: <button id="btn">Click
me</button>
let button =
document.getElementById("btn");
button.addEventListener("click", function() {
  alert("Button clicked!");
});
```

10. Asynchronous JavaScript (Promises)

```
function fetchData() {
  return new Promise((resolve, reject) => {
    // Simulate fetching data from a server
    setTimeout(() => {
      let data = [1, 2, 3, 4, 5];
      resolve(data);
    }, 2000);
  });
}
```

```
fetchData().then((data) => {
  console.log(data);
}).catch((error) => {
  console.error(error);
});
```

11. AJAX with Fetch API

```
fetch('https://jsonplaceholder.typicode.com/p
osts/1')
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => console.error(error));
```

12. ES6 Arrow Functions

```
let add = (a, b) => a + b;
console.log(add(2, 3)); // Output: 5
```

13. ES6 Classes

```
class Person {
  constructor(name, age) {
    this.name = name;
    this.age = age;
  }
}
```

```
greet() {
```

```
  return `Hello, my name is ${this.name}
and I am ${this.age} years old.`;
}
```

```
let person = new Person("Alice", 25);
console.log(person.greet()); // Output:
"Hello, my name is Alice and I am 25 years
old."
```

14. Local Storage

```
// Store data
localStorage.setItem("username", "John");
```

```
// Retrieve data
let username =
localStorage.getItem("username");
console.log(username); // Output: "John"
```

```
// Remove data
localStorage.removeItem("username");
```

15. Regular Expressions

```
let str = "Hello, my email is
john@example.com";
let pattern = /\w.-]+@[a-z]+\.[a-z]+/;
let result = str.match(pattern);
console.log(result); // Output:
["john@example.com"]
```

16. Error Handling (Try-Catch)

```
try {
  // Code that may throw an error
  let result = 10 / 0;
} catch (error) {
  console.error("Error:", error.message);
}
```

17. ES6 Modules (Export and Import)

```
// math.js
export function add(a, b) {
  return a + b;
}

// main.js
import { add } from './math.js';
console.log(add(2, 3)); // Output: 5
```

Happy Learning !