

## Table of Contents



- [1. JavaScript Syntax and Structure](#)
  - [Hello World \(First Program\)](#)
  - [Variables and Data Types](#)
  - [Comments](#)
- [2. Data Structures](#)
  - [Arrays](#)
  - [Objects \(Key-Value Pairs\)](#)
  - [Sets \(Unique Values\)](#)
- [3. Conditional Statements](#)
  - [Multiple Conditions](#)
- [4. Loops](#)
  - [For Loop](#)
  - [While Loop](#)
- [5. Functions](#)
  - [Arrow Functions](#)
- [6. Classes and Objects](#)
- [7. Exception Handling](#)
- [8. DOM Manipulation](#)
  - [Select and Modify HTML Elements](#)
- [9. Events](#)
- [10. Importing and Exporting Modules](#)
  - [Export \(in module.js\)](#)
  - [Import \(in main.js\)](#)
- [11. Array Methods](#)
- [12. Useful Built-in Functions](#)
- [13. Useful Tips](#)

# 1. JavaScript Syntax and Structure

## Hello World (First Program)

```
console.log("Hello, World!");
```

---

## Variables and Data Types

```
let name = "Alice";      // String
const age = 25;          // Integer
let height = 5.8;        // Float
let isStudent = true;    // Boolean
```

---

## Comments

```
// This is a single-line comment
```

```
/*
This is a
multi-line comment
*/
```

---

## 2. Data Structures

### Arrays

```
let fruits = ["apple", "banana", "cherry"];
fruits.push("orange");    // Add item
fruits.pop();             // Remove last item
console.log(fruits[0]);    // Access first item
```

---

### Objects (Key-Value Pairs)

```
let person = {
  name: "Bob",
  age: 30
};
```

```
console.log(person.name);    // Access value by key
person.age = 31;             // Update value
```

---

## Sets (Unique Values)

```
let uniqueNumbers = new Set([1, 2, 3, 3, 2]);
uniqueNumbers.add(4);
```

---

## 3. Conditional Statements

```
let age = 18;
if (age >= 18) {
    console.log("Adult");
} else {
    console.log("Minor");
}
```

---

### Multiple Conditions

```
let score = 85;
if (score >= 90) {
    console.log("Grade A");
} else if (score >= 75) {
    console.log("Grade B");
} else {
    console.log("Grade C");
}
```

---

## 4. Loops

### For Loop

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

---

### While Loop

```
let count = 0;  
while (count < 5) {  
  console.log(count);  
  count++;  
}
```

---

## 5. Functions

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

---

### Arrow Functions

```
const square = (num) => num * num;  
console.log(square(5)); // 25
```

---

## 6. Classes and Objects

```
class Dog {  
  constructor(name) {  
    this.name = name;  
  }  
  bark() {  
    return "Woof!";  
  }  
}
```

```
let dog1 = new Dog("Rex");  
console.log(dog1.bark());
```

---

## 7. Exception Handling

```
try {  
  let result = 10 / 0;  
} catch (error) {  
  console.error("Error occurred!");  
} finally {  
  console.log("Operation complete.");  
}
```

---

## 8. DOM Manipulation

### Select and Modify HTML Elements

```
document.getElementById("title").textContent = "Hello, JavaScript!";  
document.querySelector(".btn").style.backgroundColor = "blue";
```

---

## 9. Events

```
document.querySelector(".btn").addEventListener("click", () => {  
    alert("Button Clicked!");  
});
```

---

## 10. Importing and Exporting Modules

### Export (in module.js)

```
export const pi = 3.14;  
export function add(a, b) {  
    return a + b;  
}
```

### Import (in main.js)

```
import { pi, add } from './module.js';  
console.log(add(5, 3)); // 8
```

---

## 11. Array Methods

```
let numbers = [1, 2, 3, 4, 5];  
let doubled = numbers.map(num => num * 2); // [2, 4, 6, 8, 10]  
let filtered = numbers.filter(num => num > 3); // [4, 5]  
let sum = numbers.reduce((acc, num) => acc + num, 0); // 15
```

---

## 12. Useful Built-in Functions

```
let text = "Hello World!";  
console.log(text.length); // Length of string  
console.log(text.toUpperCase()); // Uppercase
```

```
console.log(text.toLowerCase()); // Lowercase  
console.log(text.split(" "));    // Split string into array
```

---

## 13. Useful Tips

- Use **template literals** for string interpolation:

```
let name = "John";  
console.log(`My name is ${name}`);
```

- Use **Destructuring** to extract values from objects or arrays:

```
let {name, age} = person;  
console.log(name, age);
```

- Use **spread/rest operators**:

```
let arr1 = [1, 2, 3];  
let arr2 = [...arr1, 4, 5];  
console.log(arr2); // [1, 2, 3, 4, 5]
```

---

This cheat sheet offers a quick overview of the essential aspects of JavaScript.