JavaScript Quick Reference

Comments

```
// Single-line comment

/* Multi-line comment */
```

Literals

```
42           // Number

"Hello"      // String

'Hello'      // String

true, false  // Boolean

null         // Null value

undefined    // Undefined value
```

Declarations

```
let x;                // Declare a block-scoped variable without initializing

let x = 42;           // Declare and initialize a block-scoped variable

const y = 10;         // Declare and initialize a block-scoped constant

var z = 30;           // Declare a function-scoped variable
```

Data Types

```
let num = 42;         // Number

let str = "Hello";    // String

let bool = true;      // Boolean

let obj = {a: 1, b: 2};  // Object

let arr = [1, 2, 3];     // Array
```

```
let func = function() {}; // Function

let sym = Symbol();      // Symbol
```

Operators

```
// Arithmetic Operators

+, -, *, /, %, **


// Assignment Operators

=, +=, -=, *=, /=, %=, **=


// Comparison Operators

==, ===, !=, !==, >, >=, <, <=


// Logical Operators

&&, ||, !


// Bitwise Operators

&, |, ^, ~, <<, >>, >>>


// Other Operators

typeof, instanceof, in, delete, void
```

Control Structures

```
// Conditional Statements

if (condition) { ... }

else if (condition) { ... }
```

```
else { ... }

// Switch Statement
switch (expression) {
  case value1:
    // code
    break;
  case value2:
    // code
    break;
  default:
    // code
}

// Loops
while (condition) { ... }
do { ... } while (condition);

for (initialization; condition; increment) { ... }

for (let key in object) { ... }       // Iterate over object properties
for (let value of iterable) { ... }   // Iterate over iterable values
```

Functions

```
// Function Declaration
function name(params) {
```

```
  // code
}

// Function Expression
const name = function(params) {
  // code
};

// Arrow Function
const name = (params) => {
  // code
};

// Parameters and Arguments
function greet(name = 'Guest') {
  console.log(`Hello, ${name}!`);
}

// Rest Parameters
function sum(...numbers) {
  return numbers.reduce((total, num) => total + num, 0);
}

// Returning a value
function add(a, b) {
  return a + b;
```

```
}
```

Objects

```
// Object Creation
let obj = {key1: value1, key2: value2};

// Accessing Properties
obj.key1;
obj['key2'];

// Adding/Updating Properties
obj.key3 = value3;

// Deleting Properties
delete obj.key1;
```

Arrays

```
// Array Creation
let arr = [1, 2, 3];

// Accessing Elements
arr[0];

// Adding/Removing Elements
arr.push(4);       // Add to end
arr.pop();         // Remove from end
```

```
arr.unshift(0);    // Add to beginning
arr.shift();       // Remove from beginning

// Iterating over Arrays
arr.forEach((element) => { ... });

Error Handling
try {
  // code
} catch (error) {
  // error handling
} finally {
  // always executed
}

// Throwing Errors
throw new Error('Something went wrong');

Promises and Asynchronous Programming
// Creating a Promise
let promise = new Promise((resolve, reject) => {
  // async operation
});

// Consuming a Promise
promise
```

```
  .then((result) => { ... })
  .catch((error) => { ... });

// Async/Await
async function fetchData() {
  try {
    let response = await fetch(url);
    let data = await response.json();
    // use data
  } catch (error) {
    // error handling
  }
}
```

Modules

```
// Exporting
export const name = 'value';
export function myFunction() { ... }
export default myFunction;

// Importing
import { name } from './module';
import myFunction from './module';
```

ES6 Features

```
// Let and Const
```

```
let x = 10;

const y = 20;


// Arrow Functions

const greet = (name) => `Hello, ${name}`;


// Template Literals

let message = `Hello, ${name}`;


// Destructuring

let [a, b] = [1, 2];

let {name, age} = person;


// Spread Operator

let arr = [...arr1, ...arr2];

let obj = {...obj1, ...obj2};


Standard Library Functions

// Math

Math.max(...arr);

Math.min(...arr);

Math.random();


// String Methods

str.length;

str.includes('sub');
```

```
str.indexOf('sub');

str.toUpperCase();

str.toLowerCase();


// Array Methods

arr.length;

arr.push(4);

arr.pop();

arr.unshift(0);

arr.shift();

arr.indexOf(2);

arr.includes(3);

arr.map((elem) => elem * 2);

arr.filter((elem) => elem > 2);

arr.reduce((acc, curr) => acc + curr, 0);


// Date

let date = new Date();

date.getFullYear();

date.getMonth();

date.getDate();

date.toISOString();
```