JavaScript Complete Cheatsheet

Developer Shaurya

Basics

```
console.log("Hello, World!");
alert("Hello!");
document.write("Hello JS");
```

Variables

```
var a = 10;
let b = 20;
const c = 30;
```

Data Types

```
// Primitive
let str = "Hello";
let num = 42;
let bool = true;
let undef;
let empty = null;
let big = 123n;
let sym = Symbol("id");

// Reference
let arr = [1, 2, 3];
let obj = {name: "Ravi"};
```

Operators

```
// Arithmetic: + - * / % **
// Comparison: == === != !== > < >= <=
// Logical: && || !
// Ternary: condition ? expr1 : expr2
```

Strings

```
let text = "JavaScript";
text.length;
text.toUpperCase();
text.includes("Script");
text.slice(0, 4);

// Template literals
let name = "Ravi";
```

```
`Hello, ${name}`;
```

Numbers & Math

```
Math.round(4.6);
Math.floor(4.6);
Math.ceil(4.1);
Math.random();
Math.max(1,5,9);
```

Arrays

```
let fruits = ["apple", "banana"];
fruits.push("mango");
fruits.pop();
fruits.shift();
fruits.unshift("kiwi");

// Methods
fruits.map(f => f.toUpperCase());
fruits.filter(f => f.startsWith("a"));
fruits.reduce((a,b) => a+b, 0);
```

Objects

```
let person = {
  name: "Ravi",
  age: 20,
  greet() { console.log("Hello"); }
};
console.log(person.name);
person.greet();
```

Functions

```
function add(a,b) { return a+b; }
const multiply = (a,b) => a*b;
(function(){ console.log("IIFE"); })();
```

Control Flow

```
if (x > 10) { ... }
else if (x === 10) { ... }
else { ... }

switch(day) {
  case "Mon": console.log("Start"); break;
  default: console.log("Other");
```

```
}

for (let i=0; i<5; i++) console.log(i);
while(x < 5) { x++; }
do { x++; } while(x < 5);
```

DOM Manipulation

```
document.getElementById("id");
document.querySelector(".class");

element.innerHTML = "New Text";
element.style.color = "red";

let p = document.createElement("p");
p.textContent = "Hello";
document.body.appendChild(p);
```

Events

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

ES6+ Features

```
// Destructuring
let [a,b] = [1,2];
let {name, age} = {name:"Ravi", age:20};

// Spread & Rest
let arr1 = [1,2];
let arr2 = [...arr1, 3,4];
function sum(...nums){ return nums.reduce((a,b)=>a+b); }
```

Classes & OOP

```
class Person {
  constructor(name){ this.name = name; }
  greet(){ console.log(`Hello ${this.name}`); }
}
class Student extends Person {
  constructor(name, roll){ super(name); this.roll = roll; }
}
```

Asynchronous JS

```
setTimeout(() => console.log("Done"), 1000);

let p = new Promise((res,rej)=> res("Success"));
p.then(res => console.log(res));

async function fetchData(){
  let res = await fetch("https://jsonplaceholder.typicode.com/posts/1");
  let data = await res.json();
  console.log(data);
}
```

Error Handling

```
try {
  throw new Error("Something went wrong");
} catch(e) {
  console.log(e.message);
} finally {
  console.log("Always runs");
}
```

Regex

```
let regex = /hello/i;
regex.test("Hello world");
"Hello".match(/h/i);
"123".replace(/\d/g, "#");
```

Modules

```
// export.js
export const pi = 3.14;
export default function greet(){ console.log("Hi"); }

// import.js
import greet, {pi} from "./export.js";
```

Modern Features

```
// Optional chaining
let city = user?.address?.city;

// Nullish coalescing
let value = input ?? "default";

// Private fields
class A {
  #secret = 123;
  getSecret(){ return this.#secret; }
```

```
}
```