JAVASCRIPT QUICK REFERENCE GUIDE
Introduction to Web Development Course

📘 VARIABLES & DATA TYPES

DECLARING VARIABLES

```
------------------
let name = "John";            → Modern variable (can change)
const PI = 3.14159;           → Constant (cannot change)
var oldStyle = "avoid";       → Old way (don't use)
```

DATA TYPES

```
----------
// String (text)
let message = "Hello World";
let name = 'Alice';

// Number
let age = 25;
let price = 19.99;

// Boolean
let isStudent = true;
let isLoggedIn = false;

// Array
let colors = ["red", "green", "blue"];
let numbers = [1, 2, 3, 4, 5];

// Object
let person = {
    name: "John",
    age: 30,
    email: "john@example.com"
};
```

```
// Null & Undefined
let empty = null;
let notDefined;
```

⚡ OPERATORS

ARITHMETIC

+	Addition	→ 5 + 3 = 8
-	Subtraction	→ 10 - 4 = 6
*	Multiplication	→ 6 * 2 = 12
/	Division	→ 20 / 5 = 4
%	Modulus (remainder)	→ 10 % 3 = 1
++	Increment	→ x++ (add 1)
--	Decrement	→ x-- (subtract 1)

COMPARISON

==	Equal to	→ 5 == "5" (true)
===	Strict equal	→ 5 === "5" (false)
!=	Not equal	→ 5 != 3 (true)
!==	Strict not equal	→ 5 !== "5" (true)
>	Greater than	→ 10 > 5 (true)
<	Less than	→ 3 < 7 (true)
>=	Greater or equal	→ 5 >= 5 (true)
<=	Less or equal	→ 4 <= 10 (true)

LOGICAL

&&	AND	→ (true && false) = false
||	OR	→ (true || false) = true
!	NOT	→ !true = false

🔄 CONTROL FLOW

IF STATEMENT

```
if (age >= 18) {
    console.log("Adult");
} else if (age >= 13) {
    console.log("Teenager");
} else {
    console.log("Child");
}
```

SWITCH STATEMENT

```
switch (day) {
    case "Monday":
        console.log("Start of week");
        break;
    case "Friday":
        console.log("End of week");
        break;
    default:
        console.log("Midweek");
}
```

FOR LOOP

```
for (let i = 0; i < 5; i++) {
    console.log(i);  // Prints 0, 1, 2, 3, 4
}
```

WHILE LOOP

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

```
}
```

FOR...OF LOOP (Arrays)

```
---------------------
let fruits = ["apple", "banana", "orange"];
for (let fruit of fruits) {
    console.log(fruit);
}
```

📦 FUNCTIONS

FUNCTION DECLARATION

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

let message = greet("Alice");  // "Hello, Alice!"
```

ARROW FUNCTION

```
-------------
const add = (a, b) => {
    return a + b;
};

// Short form (single expression)
const multiply = (a, b) => a * b;
```

FUNCTION WITH DEFAULT PARAMETERS

```
------------------------------
function welcome(name = "Guest") {
    return "Welcome, " + name;
}

welcome();          // "Welcome, Guest"
welcome("John");    // "Welcome, John"
```

📚 ARRAYS

COMMON METHODS

```
let numbers = [1, 2, 3, 4, 5];

numbers.push(6);              → Add to end: [1,2,3,4,5,6]
numbers.pop();               → Remove from end: [1,2,3,4,5]
numbers.shift();             → Remove from start: [2,3,4,5]
numbers.unshift(0);           → Add to start: [0,2,3,4,5]
numbers.length;              → Get length: 5
numbers.includes(3);          → Check if exists: true
numbers.indexOf(4);           → Find index: 3
```

ARRAY ITERATION

```
let nums = [1, 2, 3, 4, 5];

// forEach - loop through
nums.forEach(num => console.log(num));

// map - transform array
let doubled = nums.map(num => num * 2);  // [2,4,6,8,10]

// filter - filter array
let evens = nums.filter(num => num % 2 === 0);  // [2,4]

// reduce - reduce to single value
let sum = nums.reduce((total, num) => total + num, 0);  // 15
```

🎯 DOM MANIPULATION

```
SELECTING ELEMENTS
------------------
document.getElementById("myId");
document.querySelector(".myClass");
document.querySelectorAll("p");
document.getElementsByClassName("box");
document.getElementsByTagName("div");

CHANGING CONTENT
----------------
element.textContent = "New text";
element.innerHTML = "<strong>Bold text</strong>";
element.value = "Input value";

CHANGING STYLES
---------------
element.style.color = "red";
element.style.fontSize = "20px";
element.style.display = "none";

ADDING/REMOVING CLASSES
-----------------------
element.classList.add("active");
element.classList.remove("hidden");
element.classList.toggle("open");
element.classList.contains("selected");

EVENT LISTENERS
---------------
// Click event
button.addEventListener("click", function() {
    alert("Button clicked!");
});
```

```
// Input event
input.addEventListener("input", (e) => {
    console.log(e.target.value);
});

// Form submit
form.addEventListener("submit", (e) => {
    e.preventDefault();  // Prevent page reload
    // Handle form data
});

CREATING ELEMENTS
-----------------
let newDiv = document.createElement("div");
newDiv.textContent = "Hello";
newDiv.classList.add("box");
document.body.appendChild(newDiv);

═══════════════════════════════════════════════════════════════
═══════════════════════════════════════════

💡 COMMON PATTERNS
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
━━━━━━━━━━━━━━━━━━━━━━━━

TOGGLE DARK MODE
----------------
const toggleBtn = document.querySelector("#darkModeToggle");
toggleBtn.addEventListener("click", () => {
    document.body.classList.toggle("dark-mode");
});

FORM VALIDATION
---------------
const form = document.querySelector("#myForm");
```

```
form.addEventListener("submit", (e) => {
    e.preventDefault();

    const email = document.querySelector("#email").value;

    if (!email.includes("@")) {
        alert("Invalid email");
        return;
    }

    alert("Form submitted!");
});
```

SHOW/HIDE ELEMENT

```
const modal = document.querySelector("#modal");
const showBtn = document.querySelector("#showModal");
const closeBtn = document.querySelector("#closeModal");

showBtn.addEventListener("click", () => {
    modal.style.display = "block";
});

closeBtn.addEventListener("click", () => {
    modal.style.display = "none";
});
```

✨ PRO TIPS

✅ Use 'const' by default, 'let' only when value will change
✅ Use === instead of == for comparison

✅ Always use semicolons
✅ Use meaningful variable names
✅ Add comments for complex logic
✅ Use console.log() for debugging
✅ Handle errors with try/catch blocks
✅ Keep functions small and focused
✅ Use arrow functions for callbacks
✅ Avoid global variables

═══
═══

📚 PRACTICE EXERCISES

1. Create a counter that increments on button click
2. Build a to-do list (add, remove items)
3. Make a calculator (add, subtract, multiply, divide)
4. Create a color picker that changes background
5. Build a simple quiz with score tracking

═══
═══

═══
═══