Web Programming

JavaScript for Interactive Web Pages

Fenerbahce University

MODULE 01

JavaScript in the Browser

Key learning outcomes

1. Understand how HTML, CSS, and JavaScript work together.
2. Connect JavaScript to a page and use an external app.js file.
3. Send output to the page and to the console.
4. Write readable code with meaningful comments.

STRUCTURE **STYLE** **BEHAVIOR** **CONSOLE**

Why JavaScript after HTML and CSS?

HTML

Structure

- Headings and text
- Buttons and forms
- Lists and links

CSS

Style

- Colors and spacing
- Fonts and layout
- Responsive design

JavaScript

Behavior

- Clicks and typing
- Validation and updates
- Dynamic content

How the Browser Builds an Interactive Page

First JavaScript Connection

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Welcome</h2>
3 <p>The text below comes from JavaScript.</p>
4 <p id="message"></p>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("message").innerText =
2   "JavaScript is connected."
```

Output

Welcome

The text below comes from JavaScript.

JavaScript is connected.

Preview after the page loads.

Change Page Text with a Button

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Lesson Status</h2>
3 <button id="btn">Start Lesson</button>
4 <p id="status">Waiting...</p>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 const button = document.getElementById("btn")
2 const statusText = document.getElementById("status")
3
4 button.addEventListener("click", () => {
5   statusText.innerText = "Lesson started."
6 })
```

Output

Preview after clicking Start Lesson.

Use an External app.js File

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2 id="title">External File Demo</h2>
3 <p id="line">Loading...</p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const title = document.getElementById("title")
2 const line = document.getElementById("line")
3
4 title.innerText = "JavaScript from app.js"
5 line.innerText = "Keeping JS in a separate file makes projects cleaner."
```

Output

JavaScript from app.js

Keeping JS in a separate file makes projects cleaner.

Preview after app.js loads.

Console Output + On-Page Feedback

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Console Demo</h2>
3 <button id="run">Run</button>
4 <pre id="log"></pre>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 const runButton = document.getElementById("run")
2 const log = document.getElementById("log")
3
4 runButton.addEventListener("click", () => {
5   console.log("Button clicked")
6   log.innerText = "Open DevTools → Console"
7 })
```

Output

Preview after clicking Run.

Write Values into the Page

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Student Card</h2>
3 <p id="card"></p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const name = "Ada"
2 const track = "Frontend"
3
4 document.getElementById("card").textContent =
5   name + " studies " + track + "."
```

Output

Student Card

Ada studies Frontend.

Preview after the page loads.

Comments Help Humans Read Code

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Comment Demo</h2>
3 <p id="result"></p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 // Save the lesson topic in a variable
2 const topic = "JavaScript Basics"
3
4 // Show the topic on the page
5 document.getElementById("result").innerText = topic
```

Output

Comment Demo

JavaScript Basics

The browser shows the result, not the comments.

MODULE 02

Variables & Data Types

Key learning outcomes

1. Store values with let and const.
2. Recognize strings, numbers, and booleans.
3. Combine values with concatenation and template literals.
4. Convert text input into numbers when needed.

LET | CONST | STRING | NUMBER

Store a Value with let

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Current Topic</h2>
3 <p id="topic"></p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 let currentTopic = "Variables"
2
3 document.getElementById("topic").innerText =
4   "Today: " + currentTopic
```

Output

Current Topic

Today: Variables

Preview after the page loads.

Store a Fixed Value with const

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Lab Name</h2>
3 <p id="lab"></p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const labName = "JavaScript Lab"
2
3 document.getElementById("lab").innerText =
4   "Room label: " + labName
```

Output

Lab Name

Room label: JavaScript Lab

Preview after the page loads.

String Concatenation

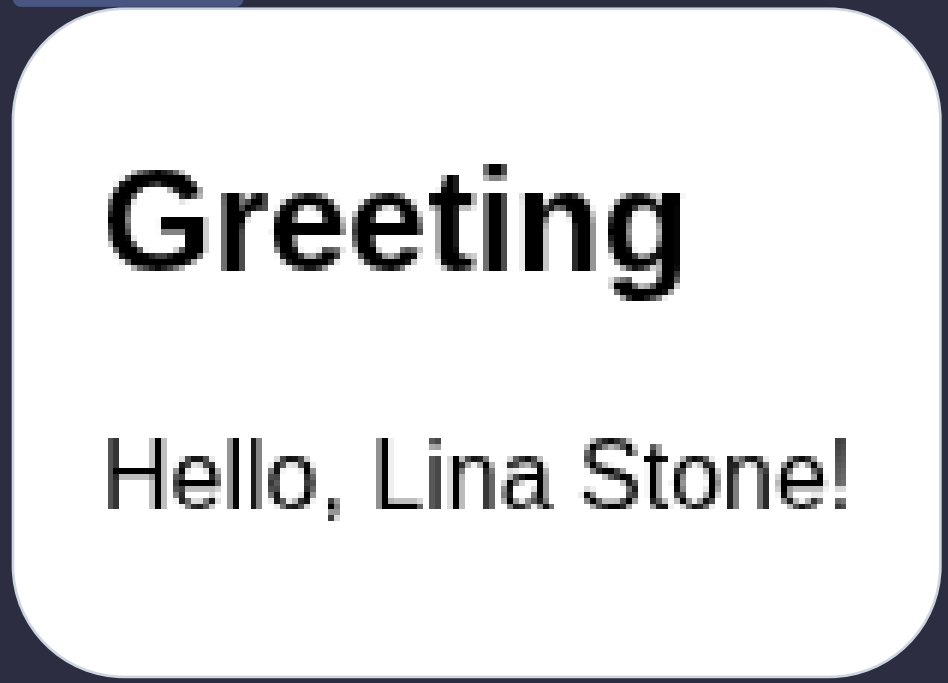
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Greeting</h2>
3 <p id="greet"></p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const firstName = "Lina"
2 const lastName = "Stone"
3
4 document.getElementById("greet").innerText =
5   "Hello, " + firstName + " " + lastName + "!"
```

Output



Preview after the page loads.

Template Literals

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Profile</h2>
3 <p id="profile"></p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const name = "Mert"
2 const age = 20
3 const city = "Istanbul"
4
5 document.getElementById("profile").innerText =
6   `${name} is ${age} years old and lives in ${city}.`
```

Output

Profile

Preview after the page loads.

Numbers and Math

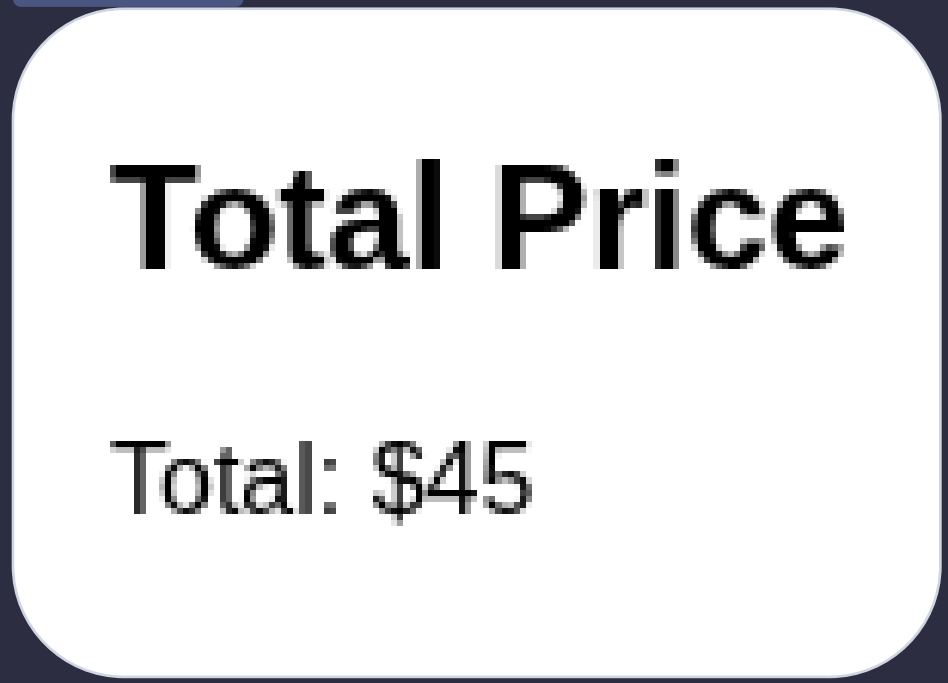
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Total Price</h2>
3 <p id="total"></p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const price = 15
2 const quantity = 3
3 const total = price * quantity
4
5 document.getElementById("total").innerText =
6   "Total: $" + total
```

Output



Preview after the page loads.

Comparison Operators

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Comparison Result</h2>
3 <p id="check"></p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const score = 80
2 const passed = score >= 60
3
4 document.getElementById("check").innerText =
5   "Passed: " + passed
```

Output

Comparison Result

Passed: true

Preview after the page loads.

Check a Data Type with typeof

HTML

```
<main style="font-family:Arial;padding:18px">
      <h2>Type Detector</h2>
<ul id="types"></ul>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
const age = 19
const user = "Lara"
const online = true

document.getElementById("types").innerHTML = `
  <li>age → ${typeof age}</li>
  <li>user → ${typeof user}</li>
  <li>online → ${typeof online}</li>
`
```

Output

Type Detector

- age → number
- user → string
- online → boolean

Preview after the page loads.

Booleans on the Page

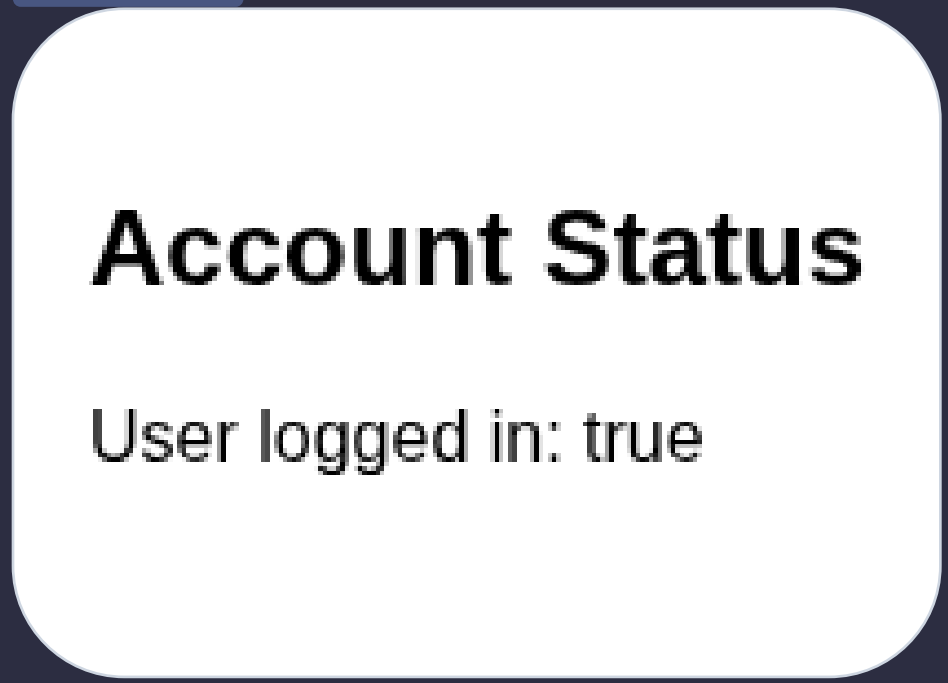
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Account Status</h2>
3 <p id="status"></p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const isLoggedIn = true
2
3 document.getElementById("status").innerText =
4   "User logged in: " + isLoggedIn
```

Output



Preview after the page loads.

Convert Text Input into Numbers

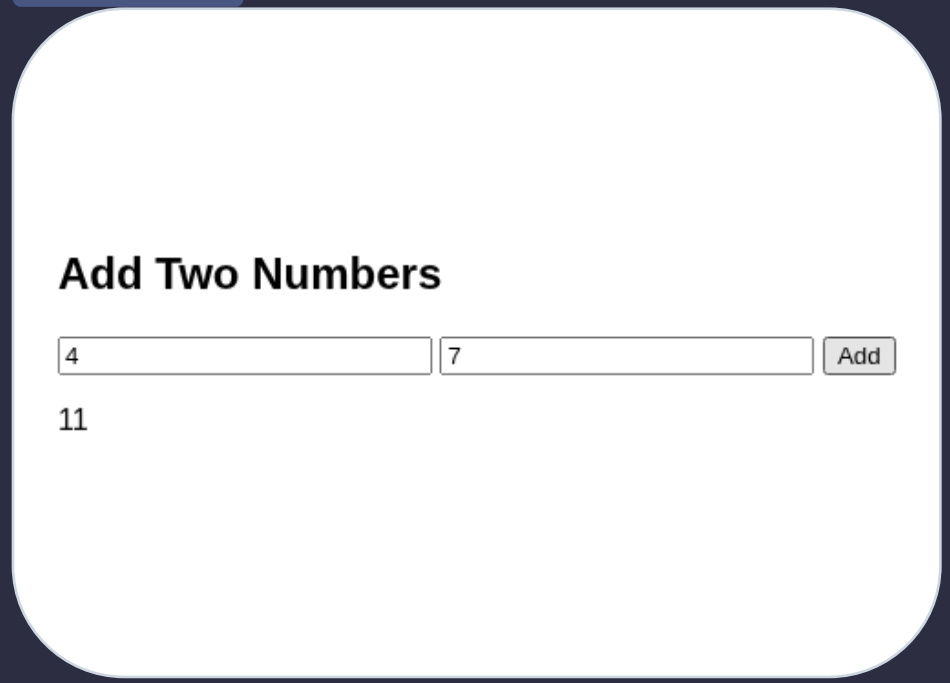
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Add Two Numbers</h2>
3 <input id="n1" value="4">
4 <input id="n2" value="7">
5 <button id="add">Add</button>
6 <p id="result"></p>
7     </main>
8     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("add").addEventListener("click", () => {
2   const n1 = Number(document.getElementById("n1").value)
3   const n2 = Number(document.getElementById("n2").value)
4   document.getElementById("result").innerText = n1 + n2
5 })
```

Output



Preview after clicking Add.

MODULE 03

Decisions with Conditions

Key learning outcomes

1. Use if, else, and else if to make decisions.
2. Compare values and react to input.
3. Combine rules with logical operators.
4. Use the ternary operator for short decisions.

IF | ELSE | LOGIC | TERNARY

if Statement: Greet the User

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Login Box</h2>
3 <input id="name" placeholder="Your name">
4 <button id="start">Start</button>
5 <p id="result"></p>
6     </main>
7     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("start").addEventListener("click", () => {
2   const name = document.getElementById("name").value
3   if (name !== "") {
4     document.getElementById("result").innerText = "Hello, " + name
5   }
6 })
```

Output

Preview after typing Ada and clicking Start.

if / else: Age Check

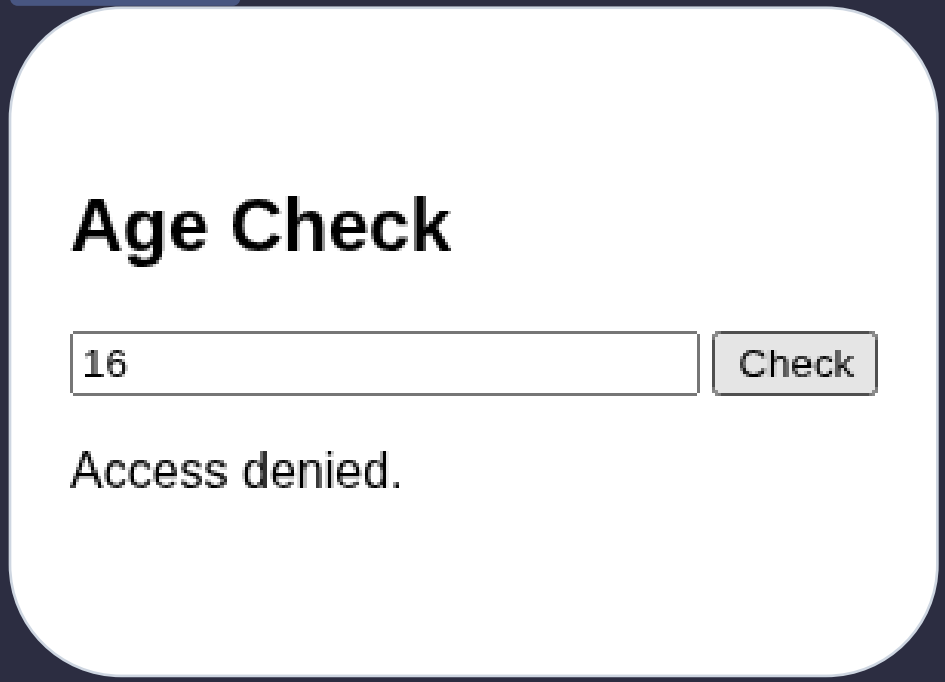
HTML

```
<main style="font-family:Arial;padding:18px">
      <h2>Age Check</h2>
<input id="age" value="16">
<button id="check">Check</button>
<p id="result"></p>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
document.getElementById("check").addEventListener("click", () => {
  const age = Number(document.getElementById("age").value)
  if (age >= 18) {
    result.innerText = "Access granted."
  } else {
    result.innerText = "Access denied."
  }
})
const result = document.getElementById("result")
```

Output



Preview after clicking Check.

else if: Grade System

HTML

```
<main style="font-family:Arial;padding:18px">
      <h2>Grade Viewer</h2>
<input id="score" value="88">
<button id="show">Show Grade</button>
<p id="grade"></p>
    </main>
    <script src="app.js"></script>
```

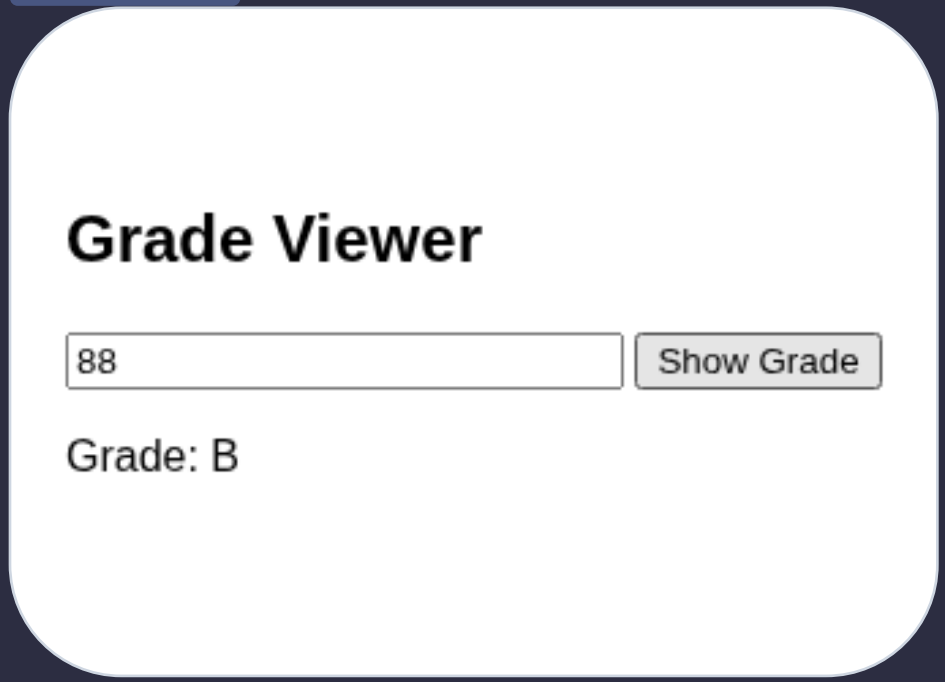
JavaScript

```
document.getElementById("show").addEventListener("click", () => {
  const score = Number(document.getElementById("score").value)
  let grade = ""

  if (score >= 90) grade = "A"
  else if (score >= 80) grade = "B"
  else if (score >= 70) grade = "C"
  else grade = "D"

  document.getElementById("grade").innerText = "Grade: " + grade
})
```

Output



Preview after clicking Show Grade.

switch: Day Planner

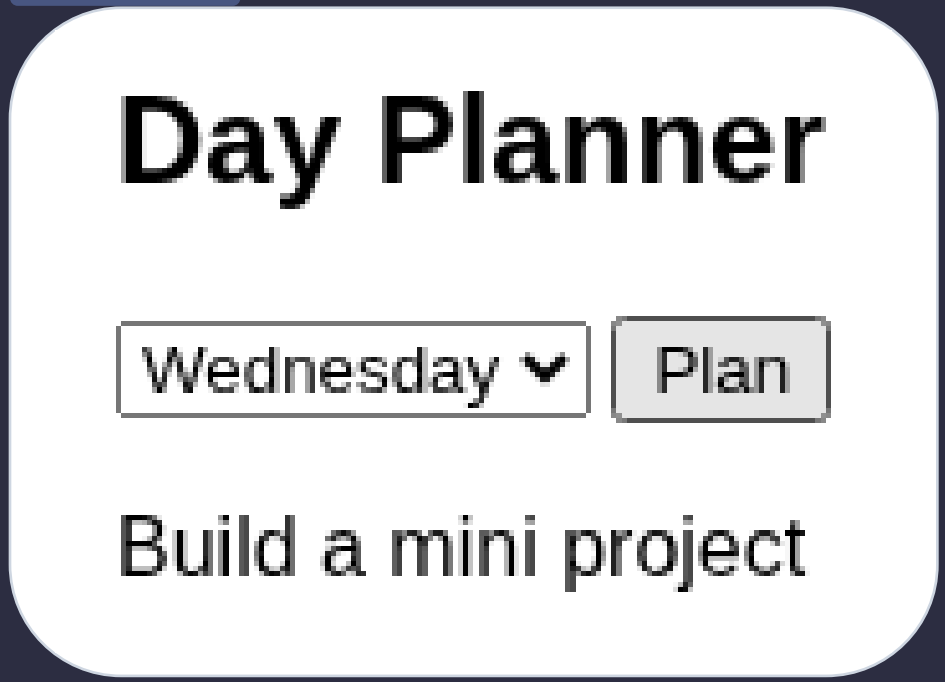
HTML

```
1 <main style="font-family:Arial;padding:18px">
 2       <h2>Day Planner</h2>
 3 <select id="day">
 4   <option>Monday</option>
 5   <option>Wednesday</option>
 6   <option>Friday</option>
 7 </select>
 8 <button id="plan">Plan</button>
 9 <p id="task"></p>
10     </main>
11     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("plan").addEventListener("click", () => {
 2   const day = document.getElementById("day").value
 3   let task = ""
 4
 5   switch (day) {
 6     case "Monday": task = "Review notes"; break
 7     case "Wednesday": task = "Build a mini project"; break
 8     default: task = "Practice exercises"
 9   }
10
11   document.getElementById("task").innerText = task
12 })
```

Output



Preview after choosing Wednesday and clicking Plan.

Logical AND (&&)

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Join the Club</h2>
3 <input id="email" placeholder="Email">
4 <label><input id="agree" type="checkbox"> I agree</label>
5 <button id="join">Join</button>
6 <p id="result"></p>
7     </main>
8     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("join").addEventListener("click", () => {
 2   const email = document.getElementById("email").value
 3   const agree = document.getElementById("agree").checked
 4
 5   if (email !== "" && agree) {
 6     result.innerText = "Ready to submit."
 7   } else {
 8     result.innerText = "Complete every requirement."
 9   }
10 })
11 const result = document.getElementById("result")
```

Output

Preview after typing an email and checking the box.

Logical OR (||)

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Display Name</h2>
3 <input id="nick" placeholder="Nickname">
4 <button id="show">Show Name</button>
5 <p id="result"></p>
6     </main>
7     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("show").addEventListener("click", () => {
2   const nick = document.getElementById("nick").value
3   document.getElementById("result").innerText = nick || "Guest"
4 })
```

Output

Preview after clicking Show Name with an empty field.

Truthy and Falsy Values

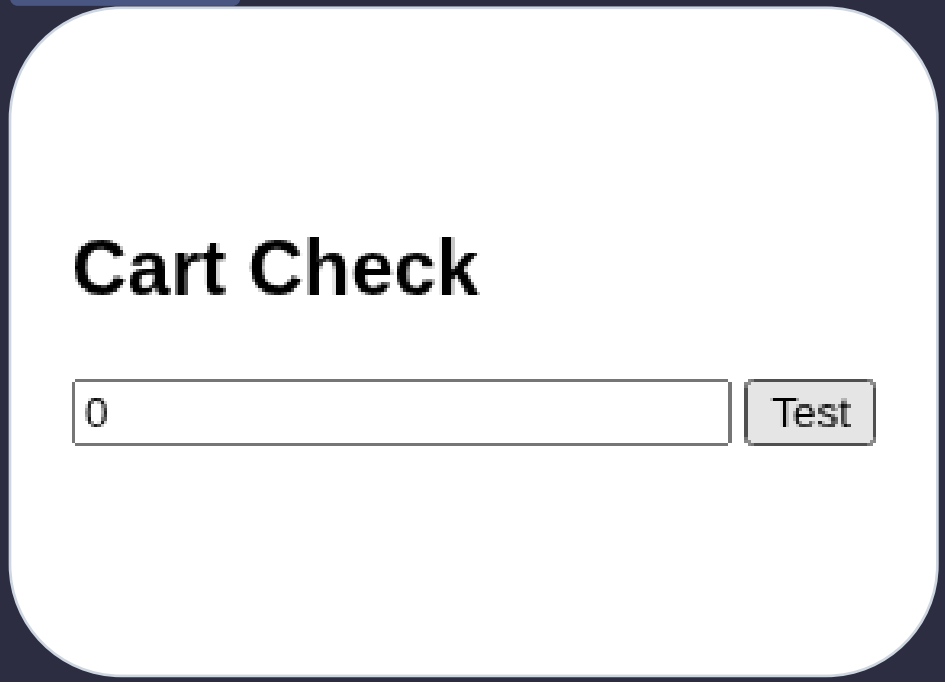
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Cart Check</h2>
3 <input id="count" value="0">
4 <button id="test">Test</button>
5 <p id="result"></p>
6     </main>
7     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("test").addEventListener("click", () => {
 2   const count = Number(document.getElementById("count").value)
 3
 4   if (count) {
 5     result.innerText = "You have items in the cart."
 6   } else {
 7     result.innerText = "The cart is empty."
 8   }
 9 })
10 const result = document.getElementById("result")
```

Output



Preview after clicking Test with 0.

Ternary Operator

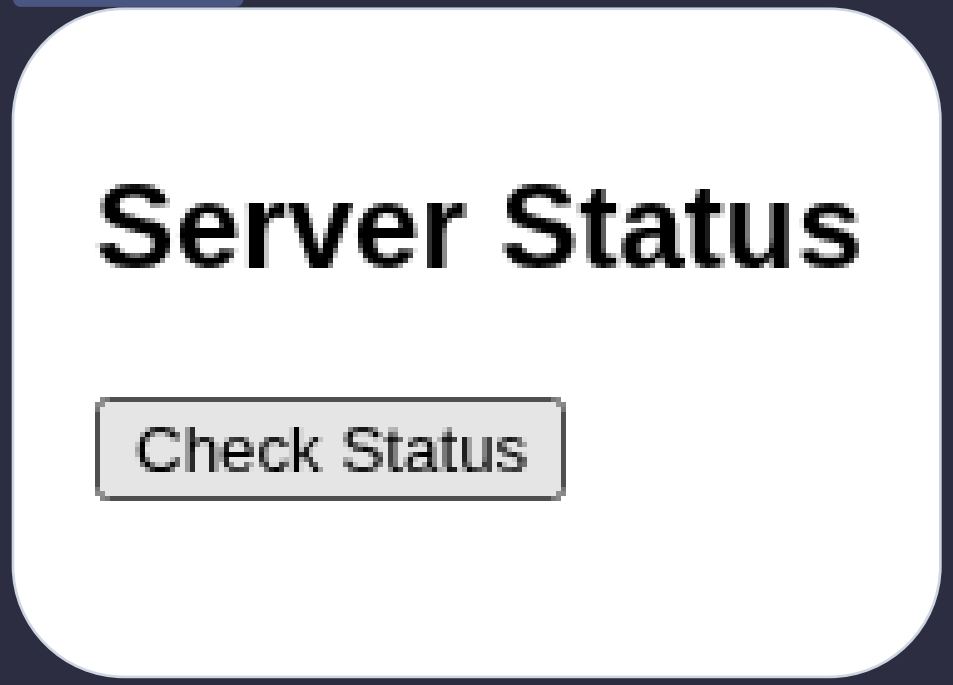
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Server Status</h2>
3 <button id="toggle">Check Status</button>
4 <p id="status"></p>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 let online = true
2
3 document.getElementById("toggle").addEventListener("click", () => {
4   document.getElementById("status").innerText =
5     online ? "Server is online." : "Server is offline."
6 })
```

Output



Preview after clicking Check Status.

Mini Rule Set: Discount Checker

HTML

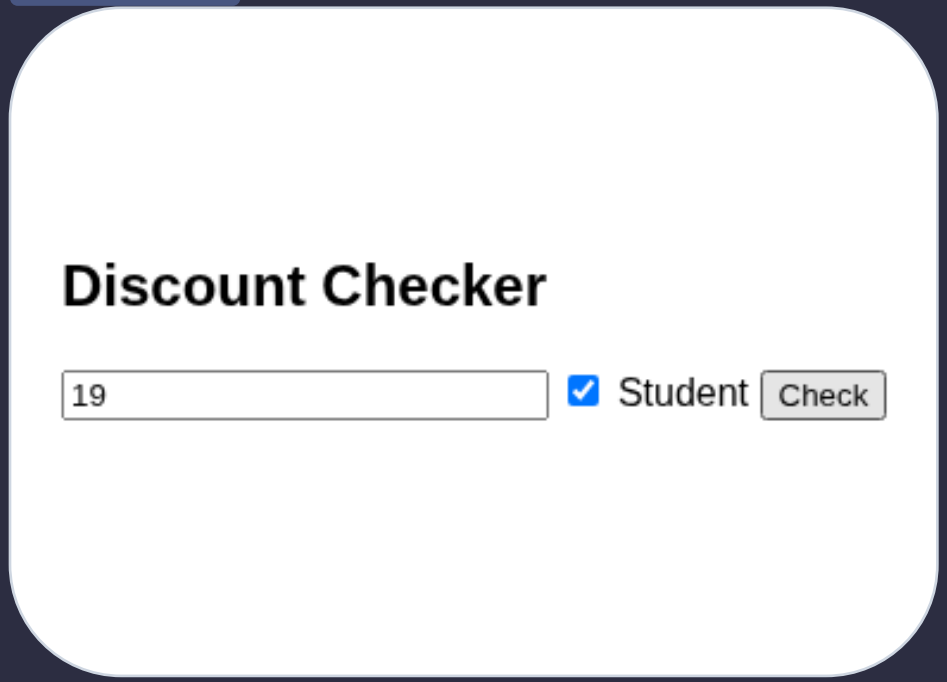
```
<main style="font-family:Arial;padding:18px">
      <h2>Discount Checker</h2>
<input id="age" value="19">
<label><input id="student" type="checkbox"> Student</label>
<button id="check">Check</button>
<p id="result"></p>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
document.getElementById("check").addEventListener("click", () => {
  const age = Number(document.getElementById("age").value)
  const isStudent = document.getElementById("student").checked

  if (age < 25 && isStudent) {
    result.innerText = "Discount unlocked."
  } else {
    result.innerText = "Standard price."
  }
})
const result = document.getElementById("result")
```

Output



Preview after checking Student and clicking Check.

MODULE 04

Functions & Reusable Logic

Key learning outcomes

1. Group code into reusable functions.
2. Pass data with parameters and get results with return.
3. Understand local and global scope.
4. Call one function many times with different values.

FUNCTION **RETURN** **SCOPE** **REUSE**

Create a Simple Function

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Message Box</h2>
3 <button id="show">Show Message</button>
4 <p id="result"></p>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 function showMessage() {
2   document.getElementById("result").innerText = "Functions group reusable steps."
3 }
4
5 document.getElementById("show").addEventListener("click", showMessage)
```

Output

Preview after clicking Show Message.

Functions with Parameters

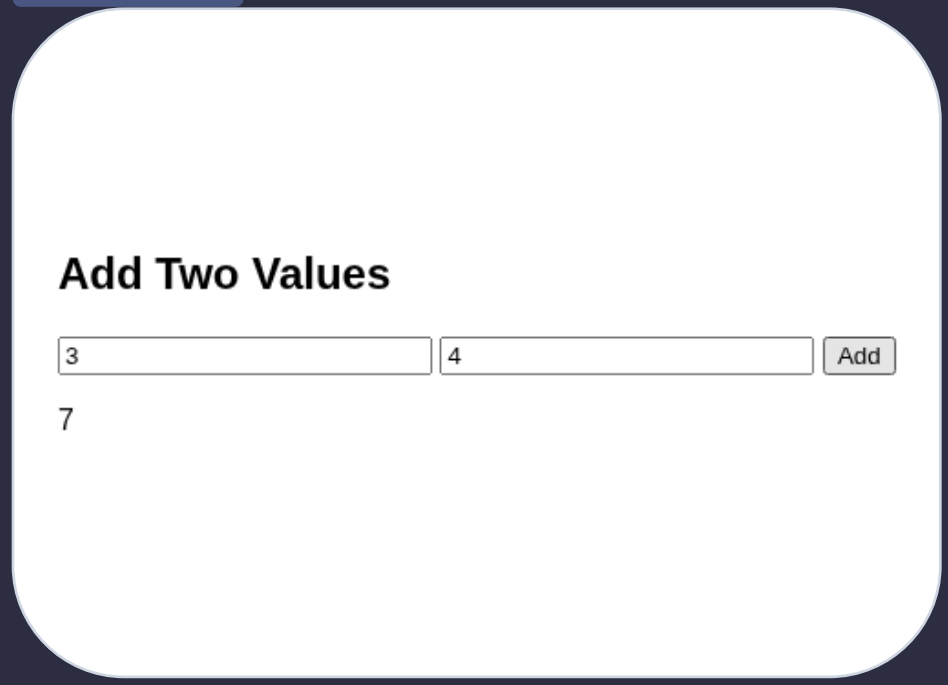
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Add Two Values</h2>
3 <input id="n1" value="3">
4 <input id="n2" value="4">
5 <button id="btn">Add</button>
6 <p id="result"></p>
7     </main>
8     <script src="app.js"></script>
```

JavaScript

```
1 function add(a, b) {
2   return a + b
3 }
4
5 document.getElementById("btn").addEventListener("click", () => {
6   const n1 = Number(document.getElementById("n1").value)
7   const n2 = Number(document.getElementById("n2").value)
8   document.getElementById("result").innerText = add(n1, n2)
9 })
```

Output



Preview after clicking Add.

Use return to Send a Result Back

HTML

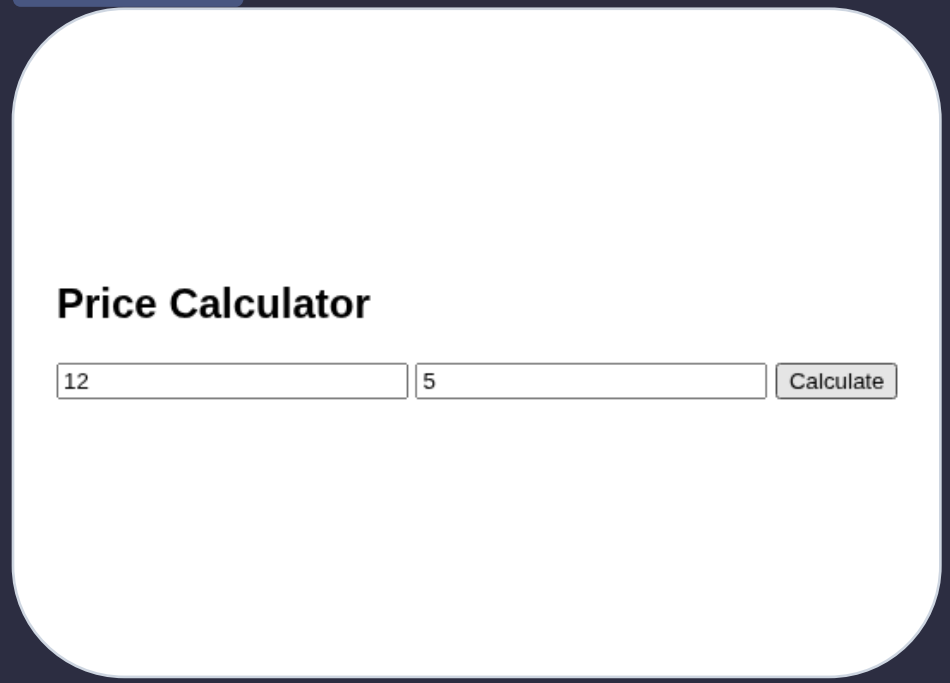
```
<main style="font-family:Arial;padding:18px">
      <h2>Price Calculator</h2>
<input id="price" value="12">
<input id="qty" value="5">
<button id="total">Calculate</button>
<p id="result"></p>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
function getTotal(price, qty) {
  return price * qty
}

document.getElementById("total").addEventListener("click", () => {
  const price = Number(document.getElementById("price").value)
  const qty = Number(document.getElementById("qty").value)
  result.innerText = "$" + getTotal(price, qty)
})
const result = document.getElementById("result")
```

Output



Preview after clicking Calculate.

Arrow Function Syntax

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Arrow Function</h2>
3 <button id="run">Run</button>
4 <p id="result"></p>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 const multiply = (a, b) => a * b
2
3 document.getElementById("run").addEventListener("click", () => {
4   document.getElementById("result").innerText = multiply(6, 7)
5 })
```

Output

Preview after clicking Run.

Global vs Local Scope

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Scope Demo</h2>
3 <button id="show">Show Scope</button>
4 <pre id="result"></pre>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 const course = "Web Programming"
2
3 function showScope() {
4   const topic = "JavaScript"
5   document.getElementById("result").innerText =
6     "Global: " + course + "\nLocal: " + topic
7 }
8
9 document.getElementById("show").addEventListener("click", showScope)
```

Output

Preview after clicking Show Scope.

Build a Full Name with a Function

HTML

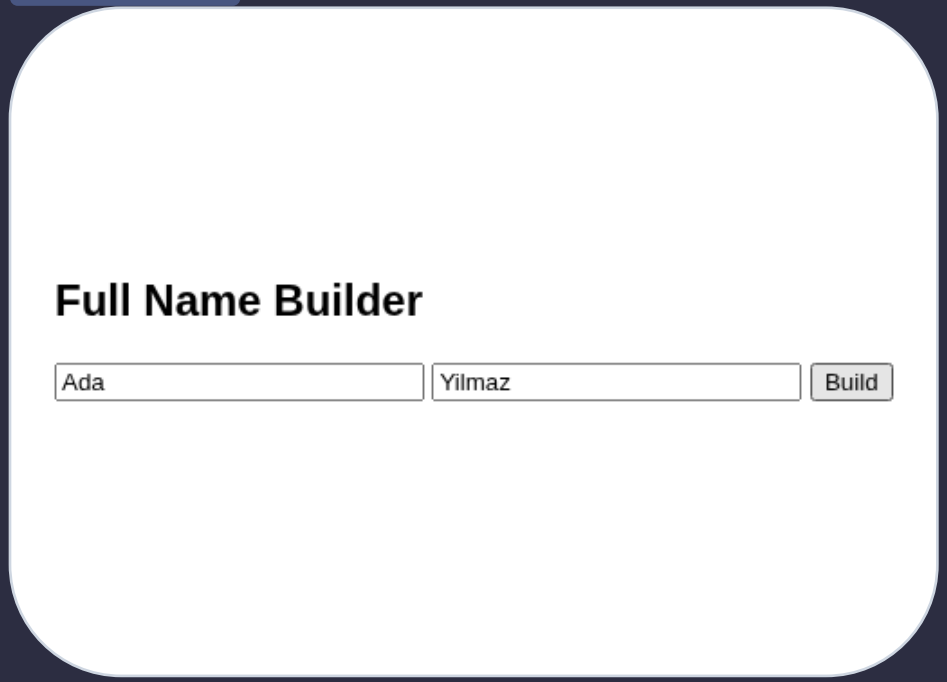
```
<main style="font-family:Arial;padding:18px">
      <h2>Full Name Builder</h2>
<input id="first" value="Ada">
<input id="last" value="Yilmaz">
<button id="make">Build</button>
<p id="result"></p>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
function fullName(first, last) {
  return `${first} ${last}`
}

document.getElementById("make").addEventListener("click", () => {
  const first = document.getElementById("first").value
  const last = document.getElementById("last").value
  result.innerText = fullName(first, last)
})
const result = document.getElementById("result")
```

Output



Preview after clicking Build.

Return HTML from a Function

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Reusable Card</h2>
3 <div id="cards"></div>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 function makeCard(name) {
2   return `<article><strong>${name}</strong><p>Active student</p></article>`
3 }
4
5 document.getElementById("cards").innerHTML =
6   makeCard("Lina")
```

Output

Reusable Card

Lina

Active student

Preview after the page loads.

Function-Powered Calculator

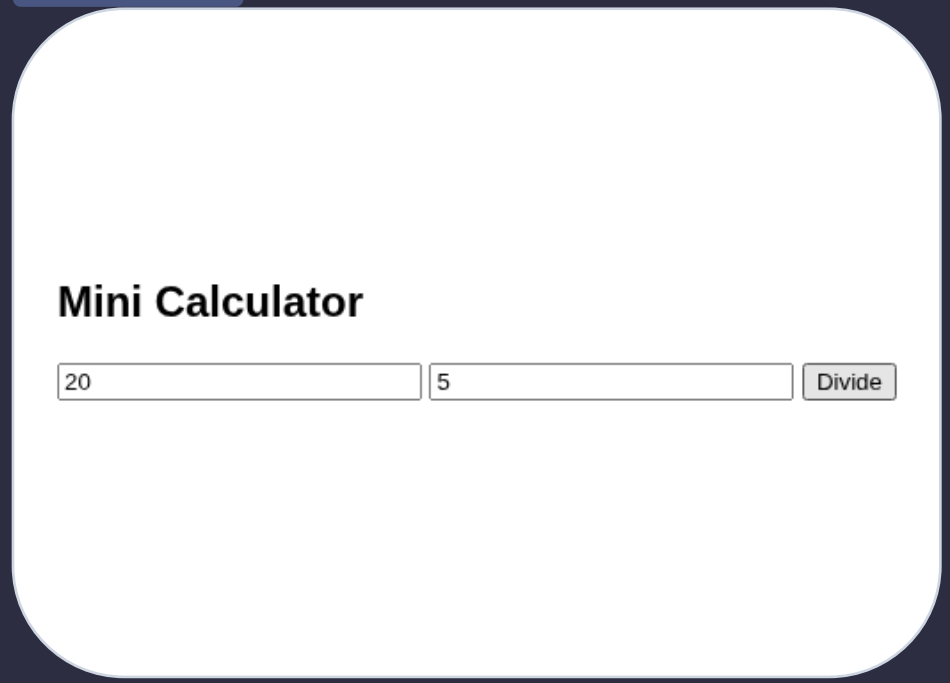
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Mini Calculator</h2>
3 <input id="a" value="20">
4 <input id="b" value="5">
5 <button id="divide">Divide</button>
6 <p id="result"></p>
7     </main>
8     <script src="app.js"></script>
```

JavaScript

```
1 function divide(a, b) {
 2   return a / b
 3 }
 4
 5 document.getElementById("divide").addEventListener("click", () => {
 6   const a = Number(document.getElementById("a").value)
 7   const b = Number(document.getElementById("b").value)
 8   result.innerText = divide(a, b)
 9 })
10 const result = document.getElementById("result")
```

Output



Preview after clicking Divide.

Receipt Line with a Function

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Receipt</h2>
3 <p id="line"></p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 function makeLine(item, qty, price) {
2   return `${item} x${qty} = $${qty * price}`
3 }
4
5 document.getElementById("line").innerText =
6   makeLine("Notebook", 2, 8)
```

Output

Receipt

Notebook x2 = $16

Preview after the page loads.

MODULE 05

Arrays & Loops

Key learning outcomes

1. Store groups of values in arrays.
2. Repeat actions with for, while, and for...of.
3. Render lists from data.
4. Stop or skip loop steps with break and continue.

ARRAY | FOR | WHILE | REPEAT

Array Basics: Render a List

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Fruit List</h2>
3 <ul id="list"></ul>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const fruits = ["Apple", "Banana", "Orange"]
2
3 document.getElementById("list").innerHTML =
4   fruits.map(fruit => `<li>${fruit}</li>`).join("")
```

Output

Fruit List

- Apple
- Banana
- Orange

Preview after the page loads.

push() and pop()

HTML

```
<main style="font-family:Arial;padding:18px">
      <h2>Editable List</h2>
<button id="add">Add Kiwi</button>
<button id="remove">Remove Last</button>
<ul id="list"></ul>
    </main>
    <script src="app.js"></script>
```

JavaScript

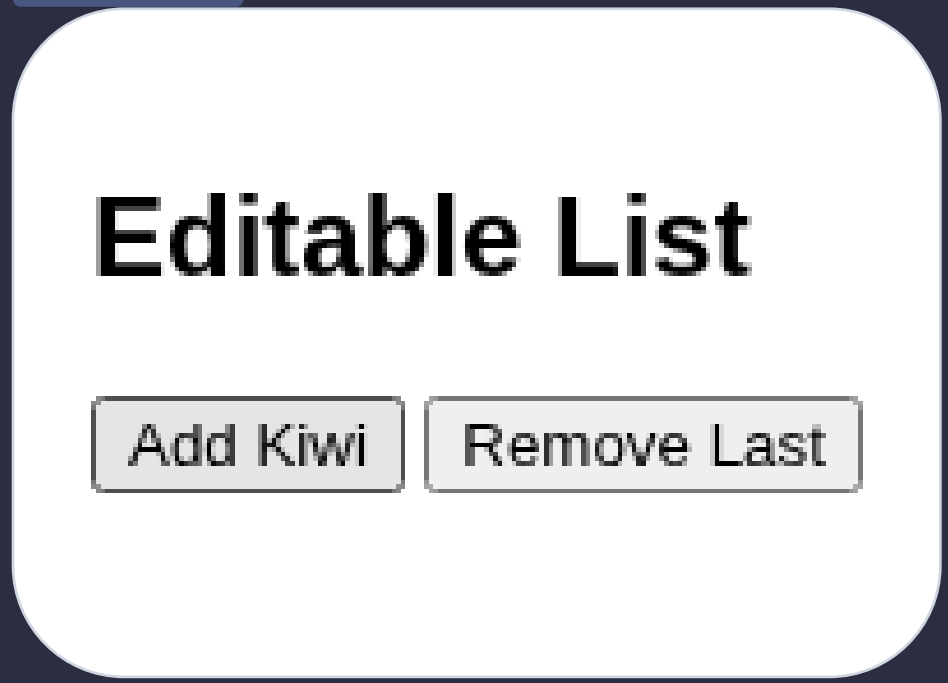
```
const fruits = ["Apple", "Banana"]

function render() {
  list.innerHTML = fruits.map(fruit => `<li>${fruit}</li>`).join("")
}

add.onclick = () => { fruits.push("Kiwi"); render() }
remove.onclick = () => { fruits.pop(); render() }

const list = document.getElementById("list")
const add = document.getElementById("add")
const remove = document.getElementById("remove")
render()
```

Output



Preview after clicking Add Kiwi.

for Loop Countdown

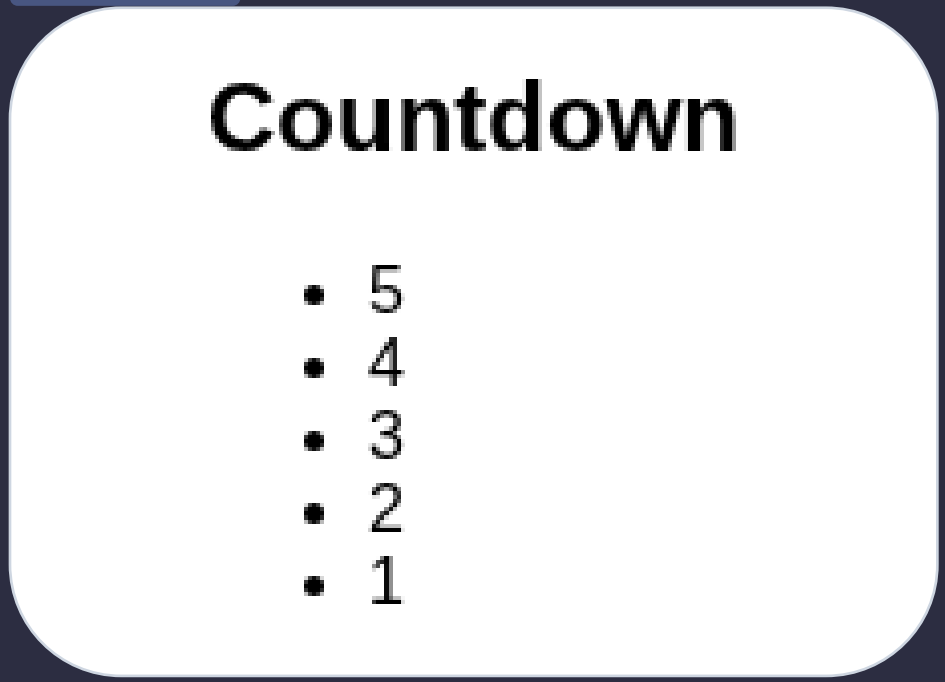
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Countdown</h2>
3 <ul id="list"></ul>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 let html = ""
2
3 for (let i = 5; i >= 1; i--) {
4   html += `<li>${i}</li>`
5 }
6
7 document.getElementById("list").innerHTML = html
```

Output



Preview after the page loads.

while Loop Counter

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>while Loop</h2>
3 <ul id="list"></ul>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 let count = 1
2 let html = ""
3
4 while (count <= 4) {
5   html += `<li>Round ${count}</li>`
6   count++
7 }
8
9 document.getElementById("list").innerHTML = html
```

Output

while Loop

Preview after the page loads.

for...of Loop

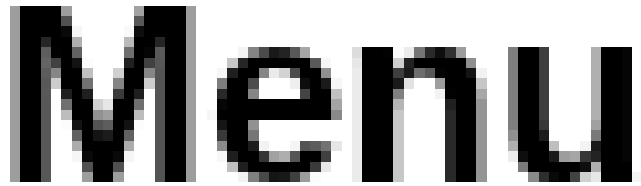
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Menu</h2>
3 <ul id="menu"></ul>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const items = ["Home", "Courses", "Projects"]
2 let html = ""
3
4 for (const item of items) {
5   html += `<li>${item}</li>`
6 }
7
8 document.getElementById("menu").innerHTML = html
```

Output



Preview after the page loads.

forEach to Render Cards

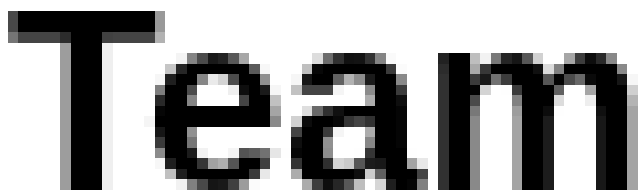
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Team</h2>
3 <div id="team"></div>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const names = ["Ada", "Lina", "Kerem"]
2 let html = ""
3
4 names.forEach(name => {
5   html += `<p>${name}</p>`
6 })
7
8 document.getElementById("team").innerHTML = html
```

Output



Preview after the page loads.

break: Stop a Loop Early

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Search Result</h2>
3 <p id="result"></p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const numbers = [2, 4, 7, 9]
 2 let found = ""
 3
 4 for (const number of numbers) {
 5   if (number === 7) {
 6     found = "Found 7"
 7     break
 8   }
 9 }
10
11 document.getElementById("result").innerText = found
```

Output

Search Result

Found 7

Preview after the page loads.

continue: Skip a Step

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Odd Numbers Only</h2>
3 <ul id="list"></ul>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 let html = ""
2
3 for (let i = 1; i <= 6; i++) {
4   if (i % 2 === 0) continue
5   html += `<li>${i}</li>`
6 }
7
8 document.getElementById("list").innerHTML = html
```

Output

Preview after the page loads.

Array-Driven Emoji Gallery

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Emoji Gallery</h2>
3 <div id="gallery"></div>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const emojis = ["😀", "🚀", "🎯", "💻"]
2
3 document.getElementById("gallery").innerHTML =
4   emojis.map(item => `<span style="font-size:32px">${item}</span>`).join(" ")
```

Output

Emoji Gallery

😀 🚀 🎯 💻

Preview after the page loads.

MODULE 06

Objects & DOM Selection

Key learning outcomes

1. Store related data inside objects.
2. Select elements with id, selectors, and selector lists.
3. Understand the DOM as a tree of elements.
4. Update text, HTML, and attributes from JavaScript.

OBJECT **DOM** **SELECTOR** **INNERHTML**

Object Basics: Student Profile

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Student Profile</h2>
3 <p id="card"></p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const student = {
2   name: "Lina",
3   major: "Software Engineering",
4   level: 2
5 }
6
7 document.getElementById("card").innerText =
8   `${student.name} • ${student.major} • Year ${student.level}`
```

Output

Student Profile

Lina • Software Engineering • Year 2

Preview after the page loads.

Array of Objects → Team Cards

HTML

```
<main style="font-family:Arial;padding:18px">
      <h2>Team Cards</h2>
<div id="team"></div>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
const team = [
  { name: "Ada", role: "HTML" },
  { name: "Kerem", role: "JavaScript" }
]

document.getElementById("team").innerHTML =
  team.map(user => `<p>${user.name} — ${user.role}</p>`).join("")
```

Output

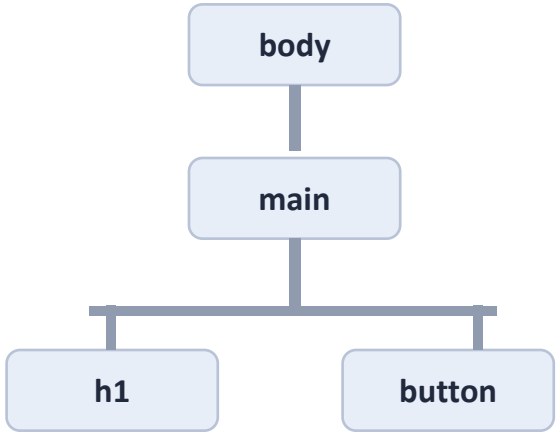
Preview after the page loads.

DOM Tree Mental Model

HTML

```
1 <body>
2   <main>
3     <h1>Title</h1>
4     <button>Click</button>
5   </main>
6 </body>
```

DOM Tree



Select One Element with getElementById()

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2 id="title">Original Title</h2>
3 <button id="change">Change Title</button>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const title = document.getElementById("title")
2
3 document.getElementById("change").addEventListener("click", () => {
4   title.innerText = "Title Changed"
5 })
```

Output

Preview after clicking Change Title.

Select with querySelector()

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <p class="note">First note</p>
3 <p class="note">Second note</p>
4 <button id="pick">Pick First</button>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("pick").addEventListener("click", () => {
2   const firstNote = document.querySelector(".note")
3   firstNote.innerText = "querySelector found me first."
4 })
```

Output

querySelector found me first.

Second note

Pick First

Preview after clicking Pick First.

Select Many with querySelectorAll()

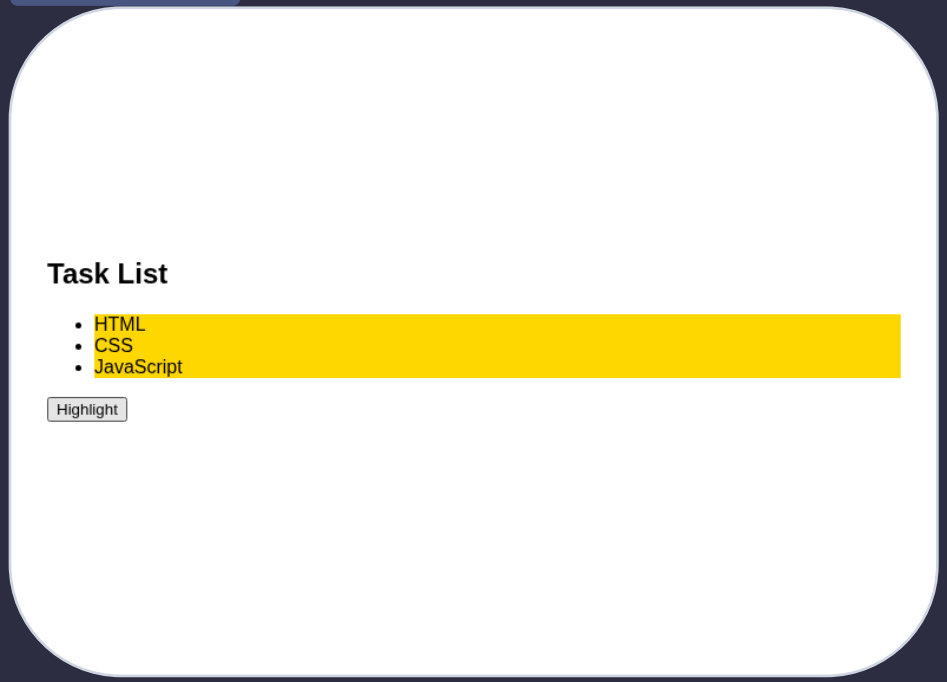
HTML

```
<main style="font-family:Arial;padding:18px">
      <h2>Task List</h2>
<ul>
  <li class="task">HTML</li>
  <li class="task">CSS</li>
  <li class="task">JavaScript</li>
</ul>
<button id="mark">Highlight</button>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
document.getElementById("mark").addEventListener("click", () => {
  const tasks = document.querySelectorAll(".task")
  tasks.forEach(task => task.style.background = "gold")
})
```

Output



Preview after clicking Highlight.

innerText Shows Plain Text

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>innerText Demo</h2>
3 <button id="show">Show</button>
4 <p id="result"></p>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("show").addEventListener("click", () => {
2   document.getElementById("result").innerText = "<strong>Hello</strong>"
3 })
```

Output

Preview after clicking Show.

innerHTML Renders HTML Tags

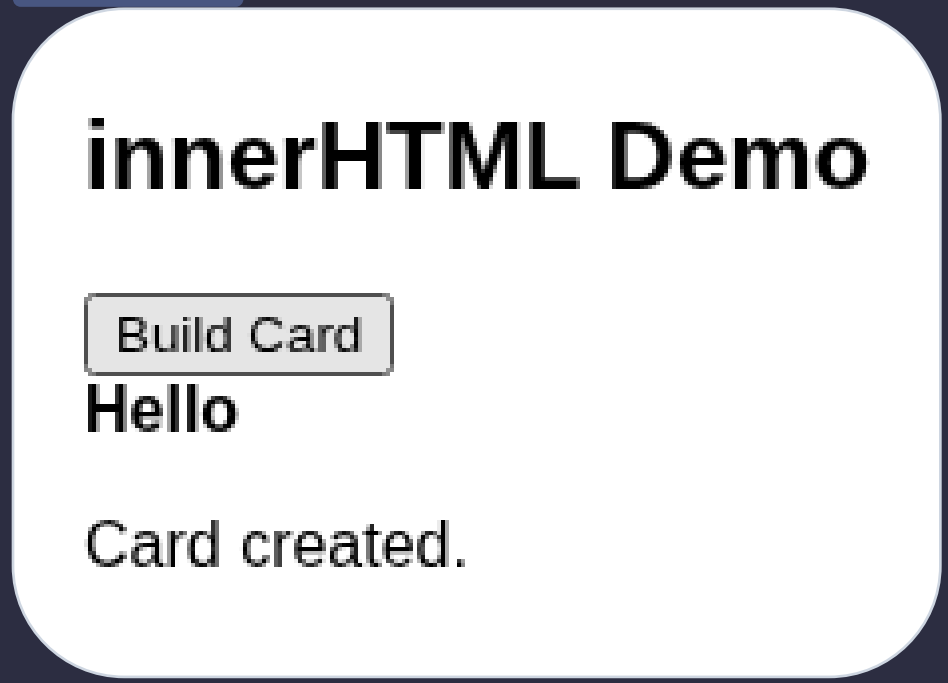
HTML

```
<main style="font-family:Arial;padding:18px">
      <h2>innerHTML Demo</h2>
<button id="build">Build Card</button>
<div id="result"></div>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
document.getElementById("build").addEventListener("click", () => {
  document.getElementById("result").innerHTML =
    "<article><strong>Hello</strong><p>Card created.</p></article>"
})
```

Output



Preview after clicking Build Card.

Change an Attribute

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Resource Link</h2>
3 <a id="link" href="#">Open resource</a>
4 <button id="update">Update Link</button>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("update").addEventListener("click", () => {
2   const link = document.getElementById("link")
3   link.href = "https://developer.mozilla.org/"
4   link.innerText = "Open MDN"
5 })
```

Output

Preview after clicking Update Link.

MODULE 07

Styling the DOM with JavaScript

Key learning outcomes

1. Change inline styles directly from JavaScript.
2. Use classList to add, remove, and toggle classes.
3. Show and hide interface states.
4. Control themes and active elements from code.

STYLE **CLASSLIST** **TOGGLE** **THEME**

Change Styles with .style

HTML

```
<main style="font-family:Arial;padding:18px">
      <h2>Color Box</h2>
<div id="box">Box</div>
<button id="paint">Paint</button>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
document.getElementById("paint").addEventListener("click", () => {
  const box = document.getElementById("box")
  box.style.color = "white"
  box.style.background = "teal"
  box.style.padding = "12px"
})
```

Output

Preview after clicking Paint.

classList.add()

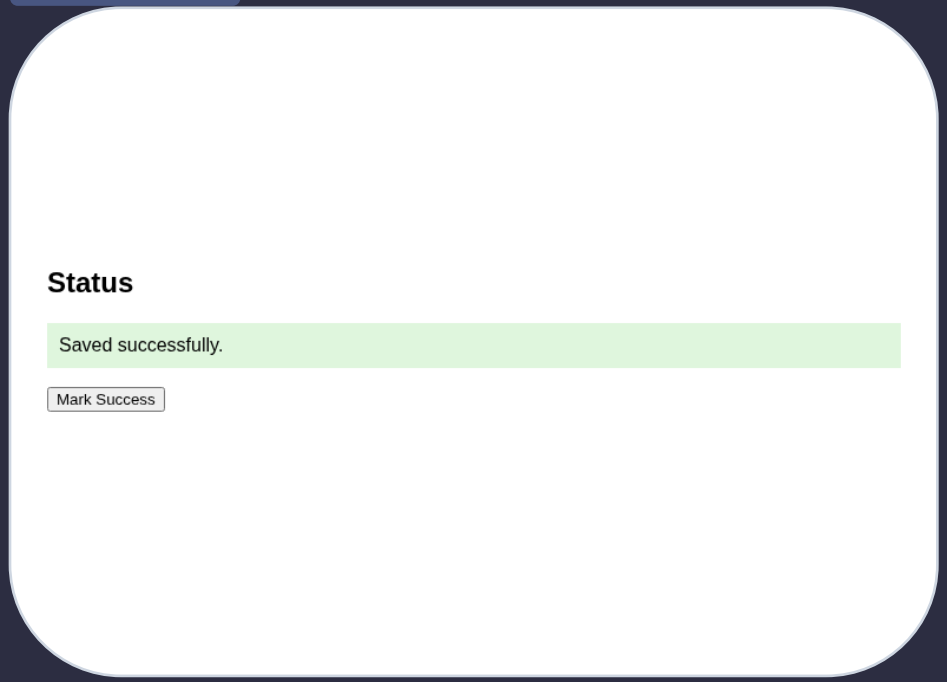
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <style>
3   .success { background: #dff6dd; padding: 10px; }
4 </style>
5 <h2>Status</h2>
6 <p id="msg">Waiting...</p>
7 <button id="ok">Mark Success</button>
8     </main>
9     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("ok").addEventListener("click", () => {
2   const msg = document.getElementById("msg")
3   msg.classList.add("success")
4   msg.innerText = "Saved successfully."
5 })
```

Output



Preview after clicking Mark Success.

classList.remove()

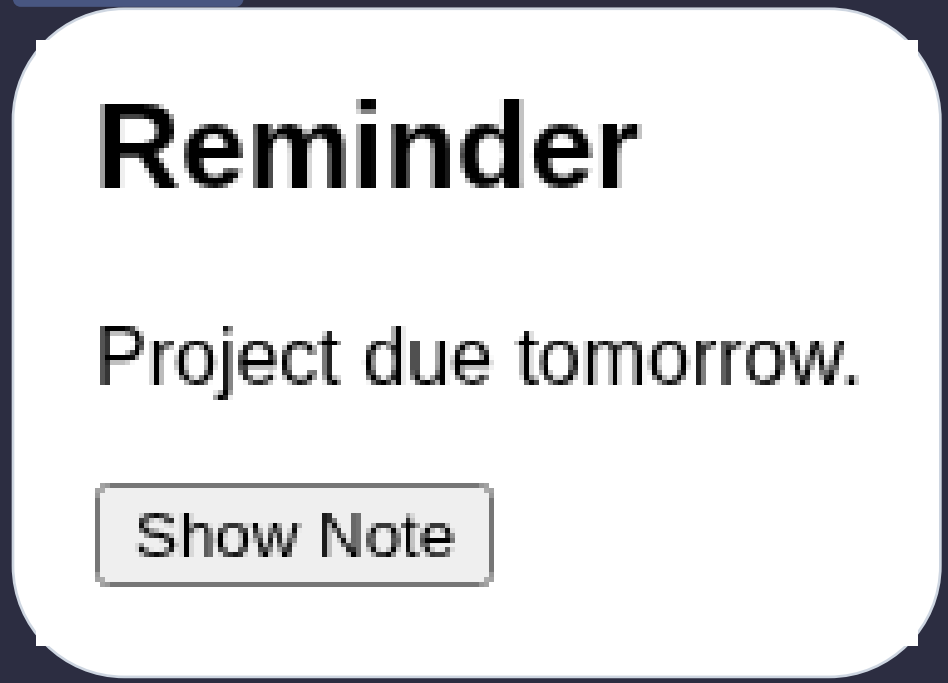
HTML

```
<main style="font-family:Arial;padding:18px">
      <style>
  .hidden { display: none; }
</style>
<h2>Reminder</h2>
<p id="note" class="hidden">Project due tomorrow.</p>
<button id="show">Show Note</button>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
document.getElementById("show").addEventListener("click", () => {
  document.getElementById("note").classList.remove("hidden")
})
```

Output



Preview after clicking Show Note.

classList.toggle()

HTML

```
<main style="font-family:Arial;padding:18px">
      <style>
  .open { display: block; }
  #answer { display: none; }
</style>
<h2>FAQ</h2>
<button id="toggle">What is DOM?</button>
<p id="answer">DOM is the browser's page tree.</p>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
document.getElementById("toggle").addEventListener("click", () => {
  document.getElementById("answer").classList.toggle("open")
})
```

Output

Preview after clicking the question.

Dark / Light Panel Toggle

HTML

```
<main style="font-family:Arial;padding:18px">
      <style>
  .dark { background: #1f2937; color: white; padding: 12px; }
</style>
<div id="panel">Theme preview</div>
<button id="switch">Toggle Theme</button>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
document.getElementById("switch").addEventListener("click", () => {
  document.getElementById("panel").classList.toggle("dark")
})
```

Output

Preview after clicking Toggle Theme.

Show / Hide Password

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Password Field</h2>
3 <input id="password" type="password" value="secret123">
4 <button id="show">Show</button>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("show").addEventListener("click", () => {
2   const password = document.getElementById("password")
3   password.type = password.type === "password" ? "text" : "password"
4 })
```

Output

Preview after clicking Show.

Active Navigation Tab

HTML

```
<main style="font-family:Arial;padding:18px">
      <style>
  .active { background: #dbeafe; }
</style>
<button class="tab">Home</button>
<button class="tab">Profile</button>
<button class="tab">Settings</button>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
const tabs = document.querySelectorAll(".tab")

tabs.forEach(tab => {
  tab.addEventListener("click", () => {
    tabs.forEach(item => item.classList.remove("active"))
    tab.classList.add("active")
  })
})
```

Output

Preview after clicking Profile.

Progress Bar Width

HTML

```
<main style="font-family:Arial;padding:18px">
      <div style="width:240px;border:1px solid #999">
  <div id="bar" style="width:30%;background:seagreen;color:white">30%</div>
</div>
<button id="grow">Grow</button>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
document.getElementById("grow").addEventListener("click", () => {
  const bar = document.getElementById("bar")
  bar.style.width = "70%"
  bar.innerText = "70%"
})
```

Output

Grow

Preview after clicking Grow.

Theme Button Text + Icon

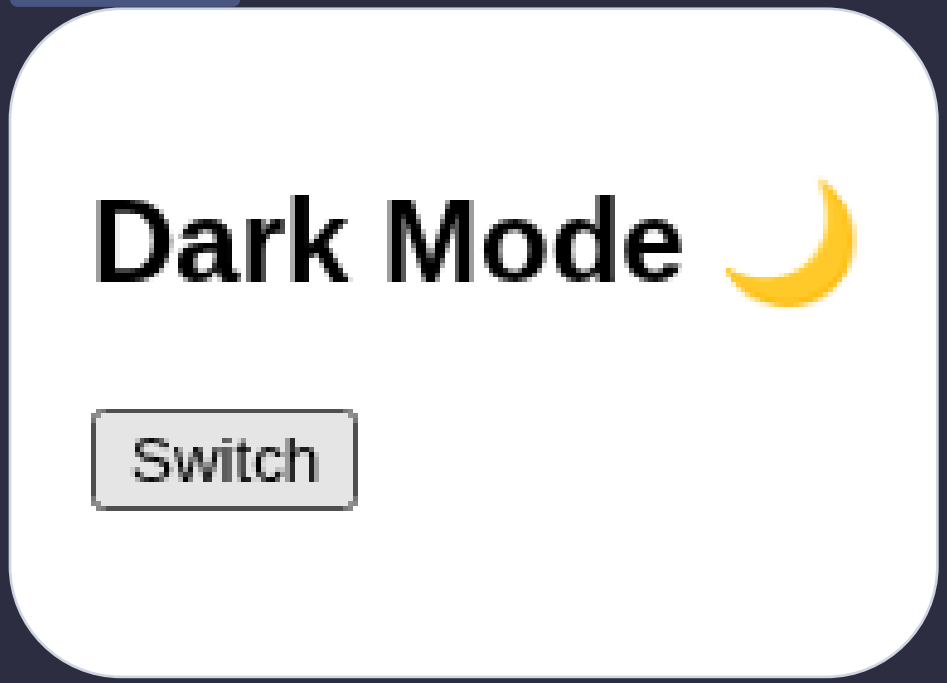
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2 id="label">Light Mode 🌞</h2>
3 <button id="toggle">Switch</button>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("toggle").addEventListener("click", () => {
2   document.getElementById("label").innerText = "Dark Mode 🌙"
3 })
```

Output



Preview after clicking Switch.

MODULE 08

Events & Forms

Key learning outcomes

1. Respond to clicks, typing, changes, and keyboard events.
2. Read values from form controls.
3. Prevent default form behavior when needed.
4. Connect events to interactive interface patterns.

CLICK | INPUT | SUBMIT | KEYBOARD

click Event: Counter

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Counter</h2>
3 <button id="plus">+</button>
4 <p id="count">0</p>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 let count = 0
2
3 document.getElementById("plus").addEventListener("click", () => {
4   count++
5   document.getElementById("count").innerText = count
6 })
```

Output

Preview after clicking + twice.

dblclick Event: Reset

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Reset Counter</h2>
3 <div id="value">5</div>
4 <button id="reset">Double-click Reset</button>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("reset").addEventListener("dblclick", () => {
2   document.getElementById("value").innerText = "0"
3 })
```

Output

Preview after double-clicking Reset.

mouseover Event

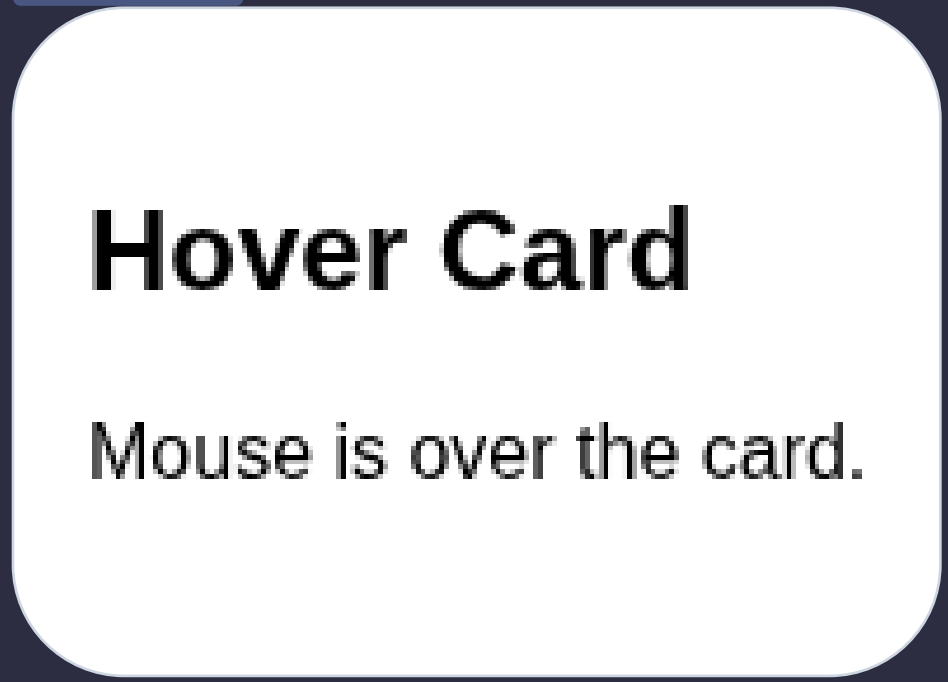
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Hover Card</h2>
3 <div id="card">Move the mouse here</div>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("card").addEventListener("mouseover", () => {
2   document.getElementById("card").innerText = "Mouse is over the card."
3 })
```

Output



Preview after moving the mouse over the card.

input Event: Live Preview

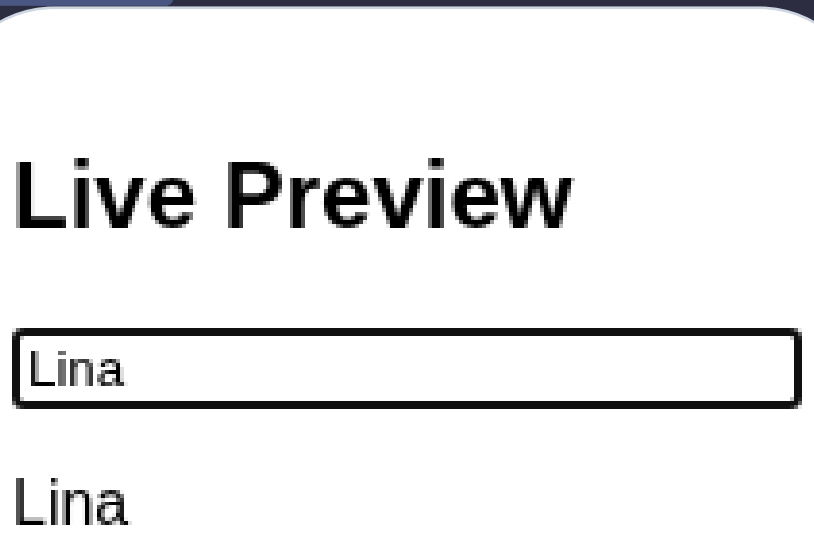
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Live Preview</h2>
3 <input id="name" placeholder="Type your name">
4 <p id="preview"></p>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("name").addEventListener("input", event => {
2   document.getElementById("preview").innerText = event.target.value
3 })
```

Output



Preview after typing Lina.

change Event: Select Menu

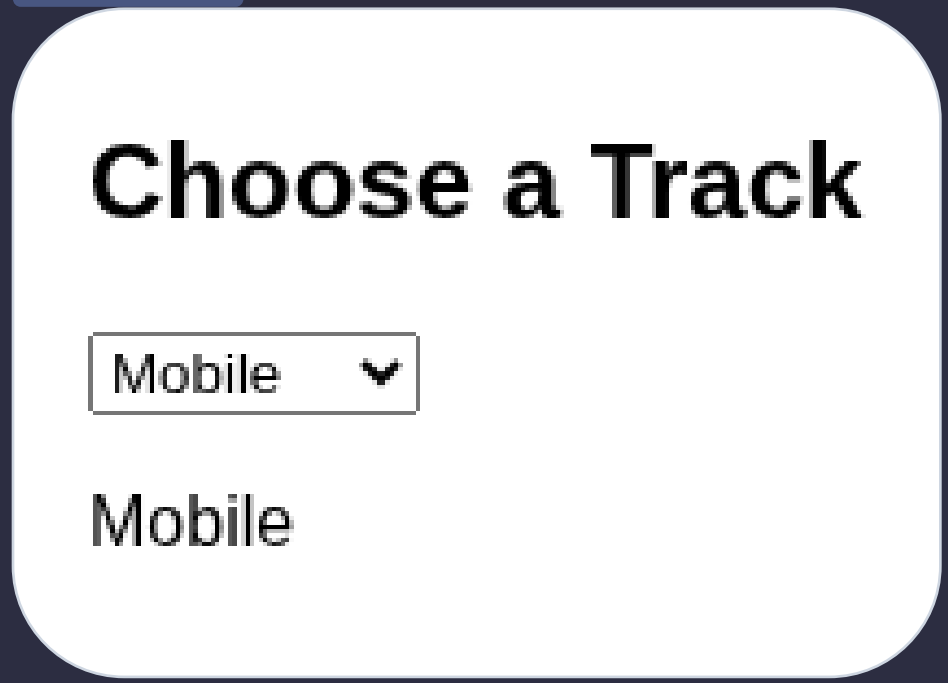
HTML

```
1 <main style="font-family:Arial;padding:18px">
 2       <h2>Choose a Track</h2>
 3 <select id="track">
 4   <option>Frontend</option>
 5   <option>Backend</option>
 6   <option>Mobile</option>
 7 </select>
 8 <p id="result"></p>
 9     </main>
10     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("track").addEventListener("change", event => {
2   document.getElementById("result").innerText = event.target.value
3 })
```

Output



Preview after changing the selection to Mobile.

submit Event: Form Handling

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Join Form</h2>
3 <form id="form">
4   <input id="email" value="student@example.com">
5   <button>Submit</button>
6 </form>
7 <p id="result"></p>
8     </main>
9     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("form").addEventListener("submit", event => {
2   event.preventDefault()
3   document.getElementById("result").innerText = "Form handled with JavaScript."
4 })
```

Output

Preview after clicking Submit.

preventDefault() in a Search Form

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Search</h2>
3 <form id="searchForm">
4   <input id="query" value="DOM">
5   <button>Search</button>
6 </form>
7 <p id="result"></p>
8     </main>
9     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("searchForm").addEventListener("submit", event => {
2   event.preventDefault()
3   const query = document.getElementById("query").value
4   document.getElementById("result").innerText = "Searching for: " + query
5 })
```

Output

Preview after clicking Search.

Read a Checkbox State

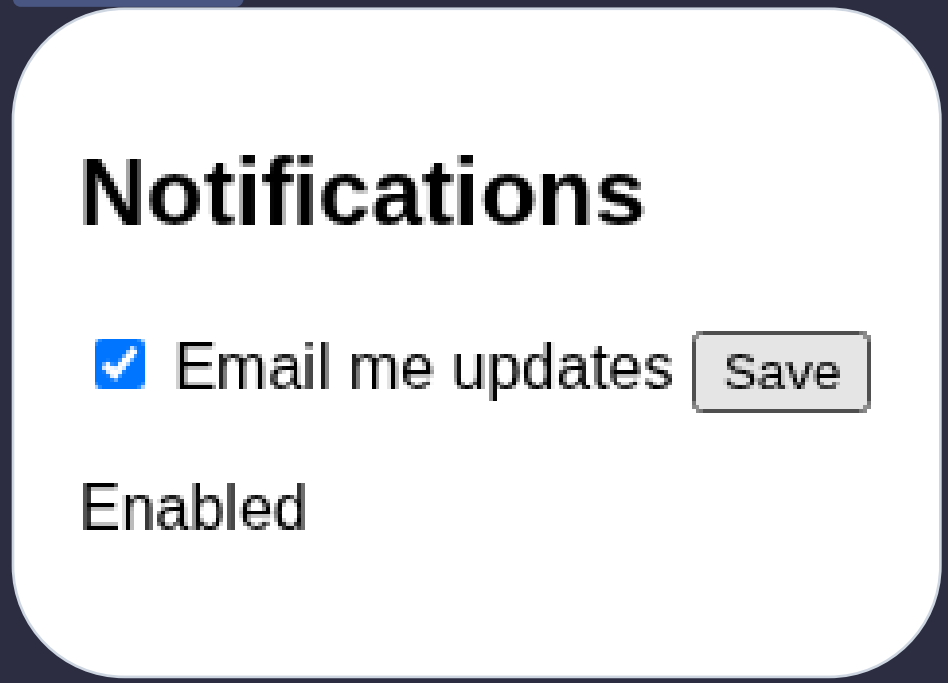
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Notifications</h2>
3 <label><input id="news" type="checkbox"> Email me updates</label>
4 <button id="save">Save</button>
5 <p id="result"></p>
6     </main>
7     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("save").addEventListener("click", () => {
2   const checked = document.getElementById("news").checked
3   document.getElementById("result").innerText = checked ? "Enabled" : "Disabled"
4 })
```

Output



Preview after checking the box and clicking Save.

keydown Event

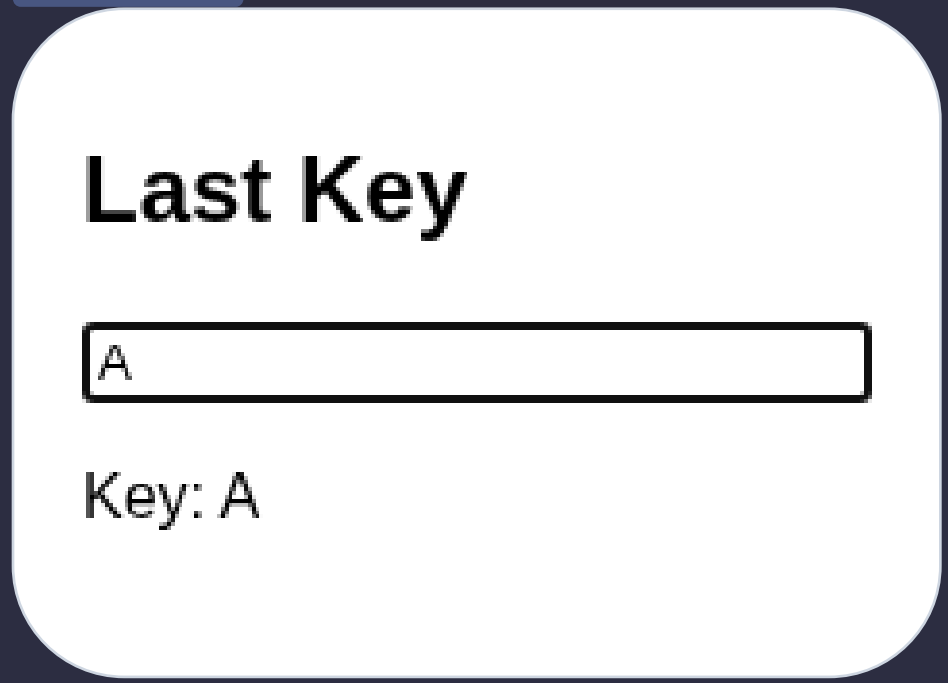
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Last Key</h2>
3 <input id="field" placeholder="Press a key">
4 <p id="result"></p>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("field").addEventListener("keydown", event => {
2   document.getElementById("result").innerText = "Key: " + event.key
3 })
```

Output



Preview after pressing A.

MODULE 09

Dynamic Elements, Timers & Storage

Key learning outcomes

1. Create and remove elements with JavaScript.
2. Store extra information with data attributes.
3. Run code later or repeatedly with timers.
4. Save simple data in localStorage and JSON.

CREATEELEMENT | TIMER | DATASET | STORAGE

Create Elements with createElement()

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Task Builder</h2>
3 <input id="taskInput" value="Review arrays">
4 <button id="add">Add Task</button>
5 <ul id="list"></ul>
6     </main>
7     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("add").addEventListener("click", () => {
2   const item = document.createElement("li")
3   item.innerText = document.getElementById("taskInput").value
4   document.getElementById("list").append(item)
5 })
```

Output

Preview after clicking Add Task.

Remove an Element

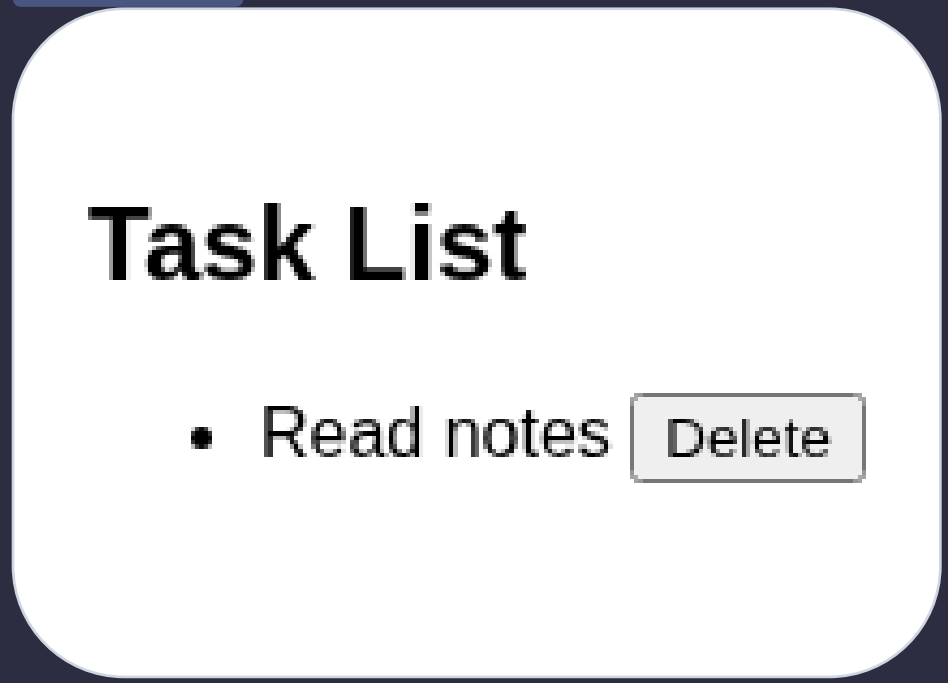
HTML

```
<main style="font-family:Arial;padding:18px">
      <h2>Task List</h2>
<ul>
  <li>Read notes <button class="delete">Delete</button></li>
  <li>Practice DOM <button class="delete">Delete</button></li>
</ul>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
document.querySelectorAll(".delete").forEach(button => {
  button.addEventListener("click", event => {
    event.target.parentElement.remove()
  })
})
```

Output



Preview after deleting the second task.

Use data-* Attributes

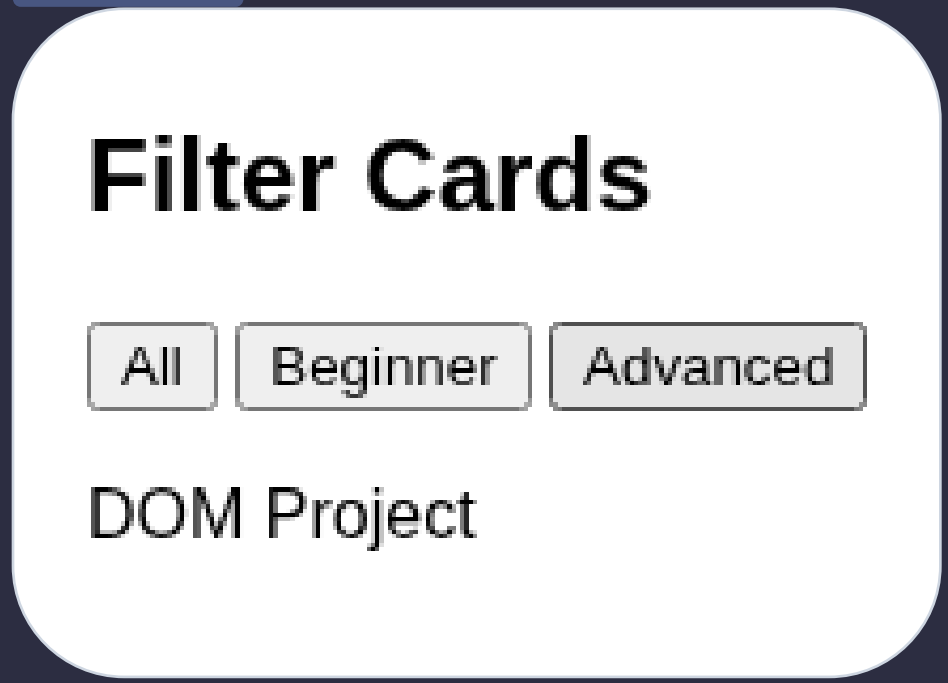
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Filter Cards</h2>
3 <button data-level="all">All</button>
4 <button data-level="beginner">Beginner</button>
5 <button data-level="advanced">Advanced</button>
6 <p class="card" data-level="beginner">HTML Basics</p>
7 <p class="card" data-level="advanced">DOM Project</p>
8     </main>
9     <script src="app.js"></script>
```

JavaScript

```
1 document.querySelectorAll("button").forEach(button => {
2   button.addEventListener("click", () => {
3     const level = button.dataset.level
4     document.querySelectorAll(".card").forEach(card => {
5       card.style.display =
6         level === "all" || card.dataset.level === level ? "block" : "none"
7     })
8   })
9 })
```

Output



Preview after clicking Advanced.

setTimeout()

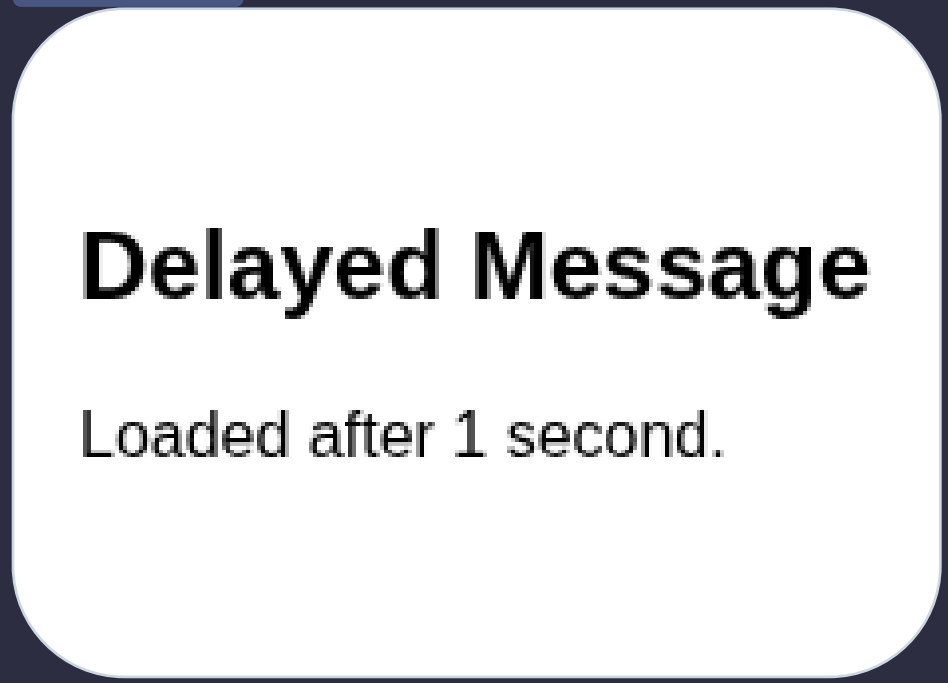
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Delayed Message</h2>
3 <p id="result">Waiting...</p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 setTimeout(() => {
2   document.getElementById("result").innerText = "Loaded after 1 second."
3 }, 1000)
```

Output



Preview after waiting 1 second.

setInterval()

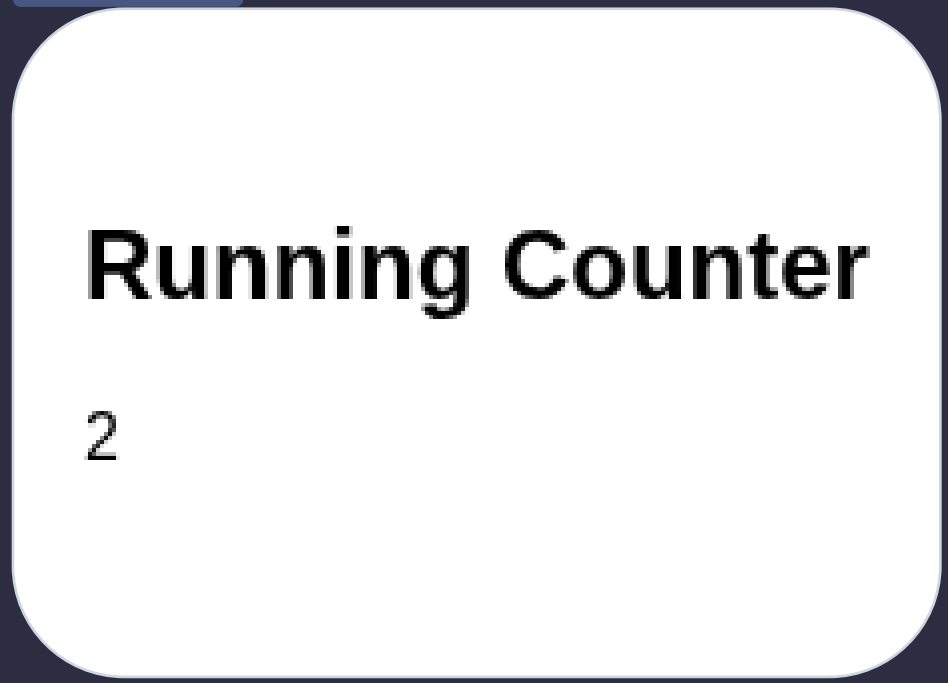
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Running Counter</h2>
3 <p id="result">0</p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 let count = 0
2
3 setInterval(() => {
4   count++
5   document.getElementById("result").innerText = count
6 }, 1000)
```

Output



Preview after about 2 seconds.

Date Object → Digital Clock

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Clock</h2>
3 <p id="clock"></p>
4     </main>
5     <script src="app.js"></script>
```

JavaScript

```
1 const date = new Date(2025, 0, 1, 14, 35, 10)
2
3 document.getElementById("clock").innerText =
4   date.toLocaleTimeString("en-GB")
```

Output

Clock

14:35:10

Preview after the page loads.

Math.random() → Dice Roller

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>Dice Roller</h2>
3 <button id="roll">Roll</button>
4 <p id="result"></p>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("roll").addEventListener("click", () => {
2   const value = Math.floor(Math.random() * 6) + 1
3   document.getElementById("result").innerText = "You rolled " + value
4 })
```

Output

Preview

Save a Value in localStorage

HTML

```
<main style="font-family:Arial;padding:18px">
      <h2>Name Saver</h2>
<input id="name" value="Lina">
<button id="save">Save</button>
<p id="result"></p>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
document.getElementById("save").addEventListener("click", () => {
  const name = document.getElementById("name").value
  localStorage.setItem("savedName", name)
  document.getElementById("result").innerText = "Saved: " + name
})
```

Output

Preview after clicking Save.

JSON.stringify() and JSON.parse()

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>JSON Demo</h2>
3 <button id="run">Convert</button>
4 <pre id="result"></pre>
5     </main>
6     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("run").addEventListener("click", () => {
2   const tasks = ["HTML", "CSS", "JavaScript"]
3   const text = JSON.stringify(tasks)
4   const back = JSON.parse(text)
5   document.getElementById("result").innerText = back.join(", ")
6 })
```

Output

Preview after clicking Convert.

MODULE 10

Mini Projects & Next Steps

Key learning outcomes

1. Combine HTML and JavaScript in small interface patterns.
2. Read, update, and create content in one project flow.
3. Practice organizing logic into clear steps.
4. Leave students with build ideas for homework.

PROJECT | TABS | QUIZ | TODO

Mini Project: Tabs

HTML

```
<main style="font-family:Arial;padding:18px">
      <style>
  .active { background: #dbeafe; }
</style>
<button class="tab" data-text="Home content">Home</button>
<button class="tab" data-text="About content">About</button>
<button class="tab" data-text="Contact content">Contact</button>
<p id="panel"></p>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
const tabs = document.querySelectorAll(".tab")
const panel = document.getElementById("panel")

tabs.forEach(tab => {
  tab.addEventListener("click", () => {
    tabs.forEach(item => item.classList.remove("active"))
    tab.classList.add("active")
    panel.innerText = tab.dataset.text
  })
})
```

Output

Preview after clicking About.

Mini Project: Accordion FAQ

HTML

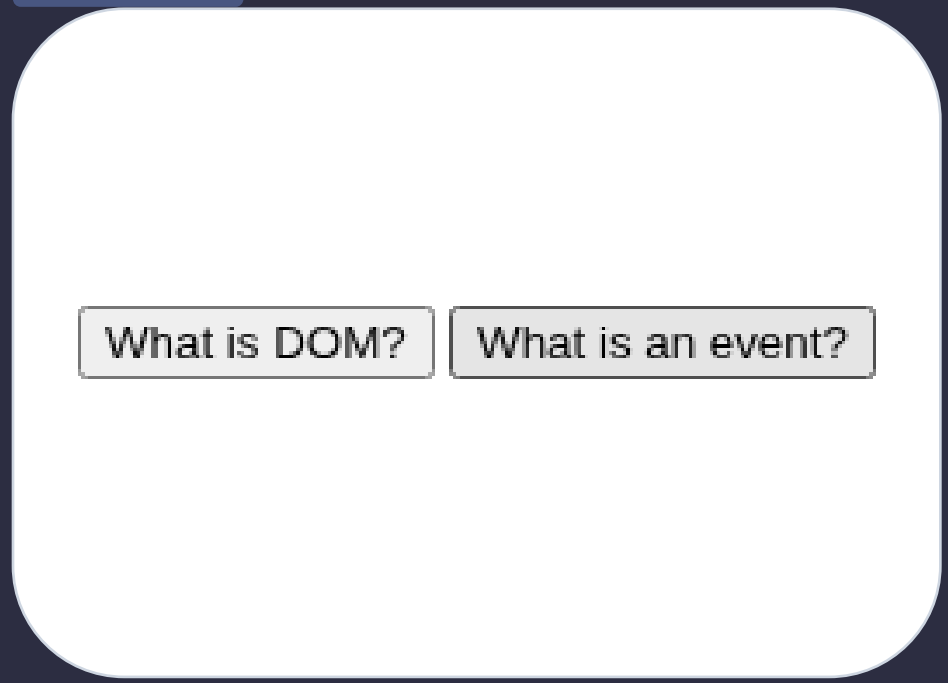
```
<main style="font-family:Arial;padding:18px">
      <style>
  .open { display: block; }
  .answer { display: none; }
</style>
<button class="question">What is DOM?</button>
<p class="answer">The document tree in the browser.</p>
<button class="question">What is an event?</button>
<p class="answer">A user or browser action.</p>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
const questions = document.querySelectorAll(".question")
const answers = document.querySelectorAll(".answer")

questions.forEach((question, index) => {
  question.addEventListener("click", () => {
    answers[index].classList.toggle("open")
  })
})
```

Output



Preview after clicking the second question.

Mini Project: Modal Window

HTML

```
<main style="font-family:Arial;padding:18px">
      <style>
  #modal { display: none; padding: 12px; border: 1px solid #999; }
  #modal.open { display: block; }
</style>
<button id="open">Open Modal</button>
<div id="modal">
  <p>Hello from the modal.</p>
  <button id="close">Close</button>
</div>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
const modal = document.getElementById("modal")

document.getElementById("open").onclick = () => modal.classList.add("open")
document.getElementById("close").onclick = () => modal.classList.remove("open")
```

Output

Preview after clicking Open Modal.

Mini Project: Quiz App

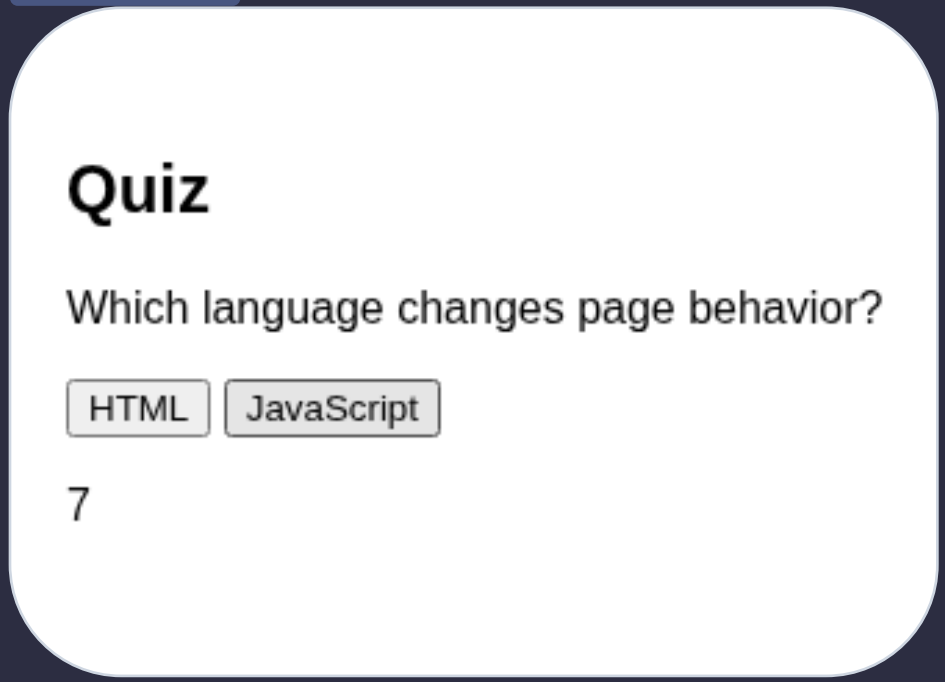
HTML

```
<main style="font-family:Arial;padding:18px">
      <h2>Quiz</h2>
<p>Which language changes page behavior?</p>
<button class="answer">HTML</button>
<button class="answer">JavaScript</button>
<p id="result"></p>
    </main>
    <script src="app.js"></script>
```

JavaScript

```
document.querySelectorAll(".answer").forEach(button => {
  button.addEventListener("click", () => {
    document.getElementById("result").innerText =
      button.innerText === "JavaScript" ? "Correct!" : "Try again."
  })
})
```

Output



Preview after choosing JavaScript.

Mini Project: Emoji Gallery

HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2 id="viewer">🌄</h2>
3 <button data-emoji="🌄">View 1</button>
4 <button data-emoji="🐶">View 2</button>
5 <button data-emoji="🚀">View 3</button>
6     </main>
7     <script src="app.js"></script>
```

JavaScript

```
1 document.querySelectorAll("button").forEach(button => {
2   button.addEventListener("click", () => {
3     document.getElementById("viewer").innerText = button.dataset.emoji
4   })
5 })
```

Output

Preview after clicking View 2.

Mini Project: Calculator

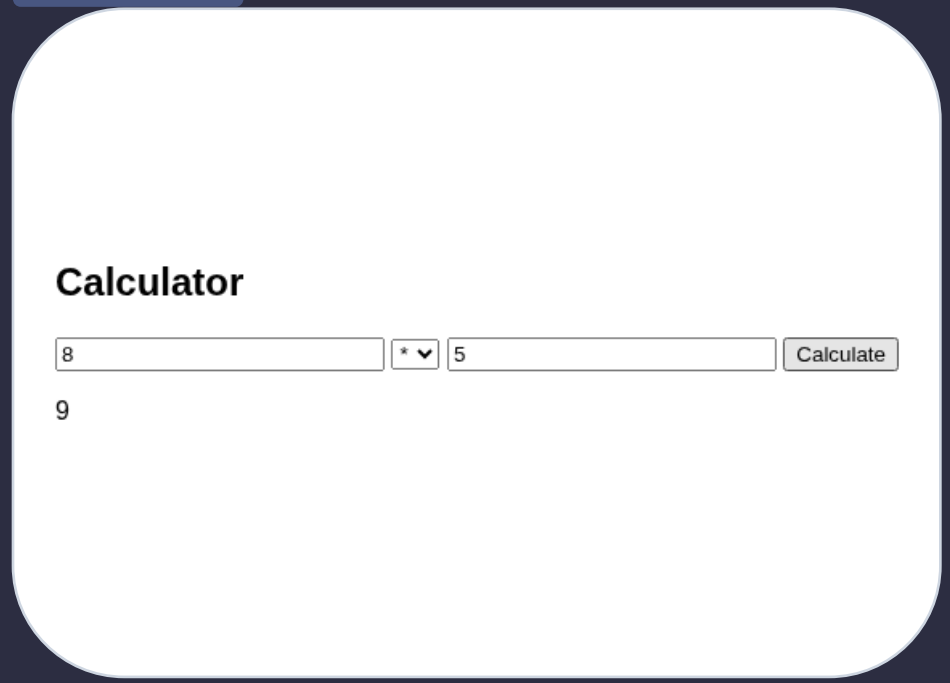
HTML

```
1 <main style="font-family:Arial;padding:18px">
 2       <h2>Calculator</h2>
 3 <input id="a" value="8">
 4 <select id="op">
 5   <option value="+">+</option>
 6   <option value="*">*</option>
 7 </select>
 8 <input id="b" value="5">
 9 <button id="calc">Calculate</button>
10 <p id="result"></p>
11     </main>
12     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("calc").addEventListener("click", () => {
2   const a = Number(document.getElementById("a").value)
3   const b = Number(document.getElementById("b").value)
4   const op = document.getElementById("op").value
5   document.getElementById("result").innerText = op === "+" ? a + b : a * b
6 })
```

Output



*Preview after choosing * and clicking Calculate.*

Mini Project: To-Do List

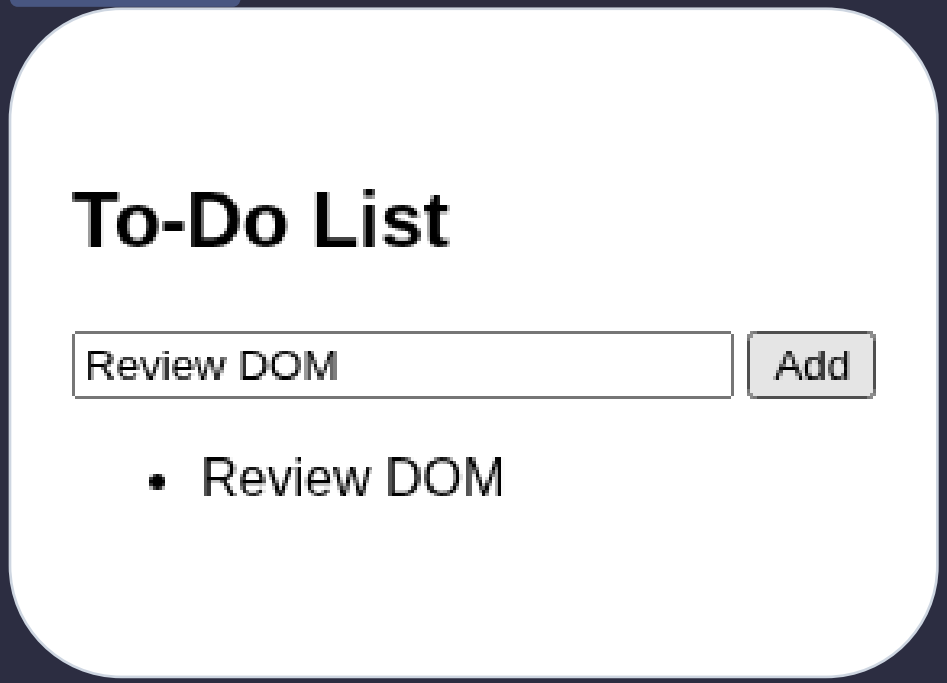
HTML

```
1 <main style="font-family:Arial;padding:18px">
2       <h2>To-Do List</h2>
3 <input id="task" value="Review DOM">
4 <button id="add">Add</button>
5 <ul id="list"></ul>
6     </main>
7     <script src="app.js"></script>
```

JavaScript

```
1 document.getElementById("add").addEventListener("click", () => {
2   const item = document.createElement("li")
3   item.innerText = document.getElementById("task").value
4   document.getElementById("list").append(item)
5 })
```

Output



Preview after clicking Add.