

Modern Intro to JavaScript — Slides

Chris Minnick

Version 1.0.0

Comprehensive course slides for “Modern Intro to JavaScript.” Page breaks occur on `#` and `##` headings; a table of contents will be auto-generated from these headings by the PDF generator.

Table of Contents

Modern Introduction to JavaScript

- Course Overview
- Welcome and Course Overview

Module 1: Course Orientation & Setup

- What "Modern JavaScript" Means in 2025
- How JavaScript Runs
- Development Environment
- Project Skeleton and Workflow
- Debugging Mindset
- Accessibility and Inclusivity
- Lab Preview

Module 2: The Language at 10,000 Feet

- Building Blocks
- Data Types and Values
- Variables and Scope
- Expressions and Operators
- Strings and Numbers
- Arrays and Objects
- Functions and Arrow Functions
- Style and Conventions
- Lab Preview

Module 3: Scaffolding with Vite (Vanilla)

- Why Vite
- Creating a Project
- Vite Anatomy
- Top-Level await
- Imports and Structure

- Build vs Preview
- Common Errors
- Lab Preview

Module 4: Core Syntax Deep Dive I

- const vs let
- Strings
- Arrays
- Objects
- Spread/Rest
- Logical Assignment
- Guarding vs Failing Fast
- Lab Preview

Module 5: Clean, Documented, and Consistent Code

- Why Consistency
- Commenting Wisely
- Effective Markdown
- Linting and Static Analysis
- Prettier vs ESLint
- JSDoc
- Hygiene Checklist
- Lab Preview

Module 6: DevTools Tour (Elements/Styles)

- Intro to DevTools
- Inspecting the DOM
- Styling Live
- Responsive Views
- Sources of Truth
- Accessibility Insights
- Pitfalls and Fixes
- Lab Preview

Module 7: DevTools (Sources/Console)

- Sources Panel
- Call Stack
- Console Essentials
- Step Debugging
- Inline Edits
- Network Preview
- Workflow Tips
- Lab Preview

Module 8: Forms & Native Validation

- Forms Bridge
- Semantic Markup
- Validation API
- Submission Handling
- Sanitizing/Parsing
- Progressive Enhancement
- UX Patterns
- Lab Preview

Module 9: CSS from a JS Perspective

- Why CSS for JS Devs
- Selector Types
- Layout Mechanics
- Flexbox/Grid Essentials
- Isolation Strategies
- Performance
- Accessibility/Visibility
- Lab Preview

Module 10: DOM & Events

- DOM Tree
- Select/Modify
- Props vs Attrs

- Events
- Handling
- A11y and Keyboard
- Data Attributes
- Lab Preview

Module 11: Classes and Modules

- Objects/Methods
- Classes
- Inheritance
- Encapsulation
- Modules Overview
- Combine Classes/Modules
- Top-Level await
- Lab Preview

Module 12: Arrays, Maps/Sets, Patterns

- Transform vs Reduce
- Immutability Mindset
- Shallow vs Deep Copies
- Iteration Patterns
- Set/Map
- Sorting and Locale
- Performance and Footguns
- Lab Preview

Module 13: Strings, Numbers, Dates, Intl

- Unicode in Practice
- Numbers and Parsing
- Intl.NumberFormat
- Intl.DateTimeFormat
- Guard Rails for Input
- Lightweight Validation
- Common Pitfalls

- Lab Preview

Module 14: Error Handling & Defensive JS

- Fundamentals
- Custom Errors
- Sync vs Async Errors
- Guard Clauses/Invariants
- Fail Soft
- Logging Discipline
- Error-First Checklist
- Lab Preview

Module 15: Program Structure & Reuse

- Public Surface
- Dependency Boundaries
- Configuration as Data
- Composition > Inheritance
- Naming & Discoverability
- Lightweight Testing
- Review Checklist
- Lab Preview

Module 16: Async I — Callbacks to Promises

- Event Loop
- Callbacks → Promises
- States and Chaining
- Combinators
- Micro vs Macro
- AbortController
- Where Promises Fail
- Lab Preview

Module 17: Async II — async/await

- Syntax & Handling

- Sequential vs Parallel
- Timeouts/Retries
- Streams Preview
- Progress & State
- Manual Testing
- Pitfalls
- Lab Preview

Module 18: fetch + JSON + CORS

- Essentials
- Status/Headers
- CORS
- POST JSON
- Retry/Backoff
- Security
- Progressive Enhancement
- Lab Preview

Module 19: URL & Forms APIs

- URLs as State
- FormData
- Serialize Filters
- Deep Links
- Accessibility
- Patterns
- To History API
- Lab Preview

Module 20: History & Storage APIs

- History Essentials
- SPA-Lite
- Storage APIs
- Privacy/Security
- Sync UI

- Persistence Patterns
- Pitfalls
- Lab Preview

Module 21: Timers, Animation, Intersection Observer

- Timers
- rAF
- Intersection Observer
- Scroll/Resize Perf
- Motion & A11y
- Measuring
- Framework Thresholds
- Lab Preview

Module 22: Organizing Small Apps

- Folders & Barrels
- API Boundary Modules
- State Without Frameworks
- DOM Update Patterns
- Error Boundaries (Conceptual)
- Build vs Ship
- Lab Preview

Module 23: Choosing & Integrating a Public API

- Picking the API
- UI Around Latency
- Pagination vs Infinite Scroll
- Error/Empty First
- Accessible Data Views
- Small CSS System
- Finish Line Checklist
- Lab Preview

Module 24: Polish & Persistence

- Search/Filter + URL
- Detail View & Routing
- Persistence
- Accessibility Sweep
- Performance Pass
- Packaging
- Optional Extras
- Lab Preview

Modern Introduction to JavaScript

ES2025 and beyond

 **Course Repository:**

<https://github.com/chrisminnick/modern-intro-to-javascript>

Version: 2.0.0

Date: October 2025

Author: Chris Minnick

Copyright © 2025 WatzThis, Inc.

All rights reserved.

Website: <https://www.watzthis.com>

Course Overview

Modern Introduction to JavaScript

What You'll Learn:

Course Structure:

Welcome and Course Overview

- Instructor intro, objectives, 5-day structure, outcomes

-

Module 1: Course Orientation & Setup

What “Modern JavaScript” Means in 2025

- ECMAScript 2025+ features; ESM everywhere; Browser vs Node

How JavaScript Runs

- Parser → engine → runtime; execution contexts; strict mode

Development Environment

- VS Code, Node, npm, Git, browsers; Vite as dev server

Project Skeleton and Workflow

- Repo structure; progressive labs; submissions

Debugging Mindset

- Break things safely; use DevTools early; intentional logging

Accessibility and Inclusivity

- A11y basics; keyboard navigation; inclusive examples

Lab Preview

- Terminal warm-up and Markdown notes

Module 2: The Language at 10,000 Feet

Building Blocks

- What makes JS unique; web platform fit; ES2025 evolution

Data Types and Values

- Primitives vs objects; typeof; dynamic typing

Variables and Scope

- let/const; block vs function scope; TDZ

Expressions and Operators

- Arithmetic, comparison, logical; == vs ===; ??, ?., ||=, &&=

Strings and Numbers

- Template literals; parsing; floating point pitfalls

Arrays and Objects

- Literals, access, update, nesting; readability

Functions and Arrow Functions

- Declarations vs expressions; lexical this; defaults

Style and Conventions

- Formatting, naming; avoid magic numbers; readability

Lab Preview

- First JS-driven page

Module 3: Scaffolding with Vite (Vanilla)

Why Vite

- HMR, ESM, fast dev; small/mid-size projects

Creating a Project

- Command; structure; npm run dev

Vite Anatomy

- index.html entry; main.js; public/; script injection

Top-Level await

- Works in modules; simple fetch example

Imports and Structure

- modules/, utils/, components/; import syntax

Build vs Preview

- `npm run build; dist/; npm run preview`

Common Errors

- Missing `type="module"`; import paths; CORS; clear cache

Lab Preview

- Scaffold, edit, HMR

Module 4: Core Syntax Deep Dive I

const vs let

- Mutation vs reassignment; immutability mindset

Strings

- Template literals; includes/startsWith/slice; formatting

Arrays

- map/filter/reduce; find/some/every; avoid mutation

Objects

- Access/update; computed keys; destructuring

Spread/Rest

- Clone/merge; collect args; pitfalls

Logical Assignment

- `||=`, `&&=`, `??=`; defaults and guards

Guarding vs Failing Fast

- Early returns; validation; simplicity

Lab Preview

- Data transformation drills

Module 5: Clean, Documented, and Consistent Code

Why Consistency

- Readability, maintainability, onboarding, fewer bugs

Commenting Wisely

- Focus on why over what; inline vs block

Effective Markdown

- Headings, lists, code blocks; install/usage sections

Linting and Static Analysis

- ESLint setup and autofix

Prettier vs ESLint

- Formatting vs style; VS Code integration

JSDoc

- Types for IntelliSense; simple example

Hygiene Checklist

- Naming, structure, format+lint before commit

Lab Preview

- ESLint + Prettier and docs

Module 6: DevTools Tour (Elements/Styles)

Intro to DevTools

- Why master DevTools; browser differences; live debugging

Inspecting the DOM

- Elements panel; DOM tree; attributes

Styling Live

- Box model; computed styles; visual debugging

Responsive Views

- Device emulation; layout constraints

Sources of Truth

- Temp vs saved changes; mapping files

Accessibility Insights

- A11y tree; contrast; keyboard nav testing

Pitfalls and Fixes

- Overlapping styles; specificity; caching

Lab Preview

- Inspect/edit a page to understand DOM+CSS

Module 7: DevTools (Sources/Console)

Sources Panel

- Find scripts; breadcrumbs; breakpoints

Call Stack

- Execution; scopes; locals vs globals

Console Essentials

- log/table/group/time; filtering

Step Debugging

- In/over/out; watch expressions; conditionals

Inline Edits

- Quick fixes; temporary changes

Network Preview

- Responses; 404s; prep for async debugging

Workflow Tips

- Start console → breakpoints; document findings

Lab Preview

- Countdown bug hunt

Module 8: Forms & Native Validation

Forms Bridge

- Built-in behavior; leverage native validation

Semantic Markup

- Label/for; inputs; fieldset/legend

Validation API

- required/pattern/min/max/step; checkValidity/reportValidity

Submission Handling

- `preventDefault`; `FormData`

Sanitizing/Parsing

- Types over strings; avoid unsafe concat

Progressive Enhancement

- Build on native; maintain a11y

UX Patterns

- Inline errors; focus management; touch/keyboard

Lab Preview

- Build a contact form

Module 9: CSS from a JS Perspective

Why CSS for JS Devs

- JS interactivity depends on selectors/classes

Selector Types

- Classes, IDs, attributes; :has/:is/:where

Layout Mechanics

- Positioning; z-index; pointer-events

Flexbox/Grid Essentials

- Containers; toggling layouts

Isolation Strategies

- BEM; modules; Shadow DOM preview

Performance

- Avoid forced reflows; rAF; batching

Accessibility/Visibility

- `display:none` vs `visibility:hidden`; `aria-hidden`

Lab Preview

- Dynamic styling with selectors

Module 10: DOM & Events

DOM Tree

- API for HTML; nodes vs elements

Select/Modify

- `querySelector`, `createElement`, `append`, `remove`

Props vs Attrs

- Differences; sync issues; toggling states

Events

- Capture→target→bubble; delegation; common events

Handling

- `addEventListener`; `remove`; event object

A11y and Keyboard

- Click + keyboard support; focus order

Data Attributes

- dataset for configuration

Lab Preview

- Interactive page with events and DOM updates

Module 11: Classes and Modules

Objects/Methods

- this in functions vs arrows; method patterns

Classes

- Syntax; constructors; instance vs static

Inheritance

- extends; super; overrides

Encapsulation

- #private; getters/setters

Modules Overview

- import/export; project structure

Combine Classes/Modules

- Split definitions; clean imports

Top-Level await

- Load config before start; execution order

Lab Preview

- Convert utilities to classes/modules

Module 12: Arrays, Maps/Sets, Patterns

Transform vs Reduce

- Choose the right tool; purity; readability

Immutability Mindset

- Avoid in-place mutation; copy patterns

Shallow vs Deep Copies

- spread/Object.assign vs structuredClone

Iteration Patterns

- `for...of`, `forEach`, iterators; `some`/`every`

Set/Map

- Membership, dedupe, indices

Sorting and Locale

- Intl.Collator; stable sorts; comparators

Performance and Footguns

- sort/reverse mutate; measure before optimizing

Lab Preview

- Dedupe/index/sort workflow

Module 13: Strings, Numbers, Dates, Intl

Unicode in Practice

- `normalize`; `codePointAt`; practical comparison

Numbers and Parsing

- Number vs parseInt/parseFloat; Number.isNaN; BigInt

Intl.NumberFormat

- Currency/percent/compact; format at the edge

Intl.DateTimeFormat

- Locale-friendly dates; input/output tips

Guard Rails for Input

- Validate/coerce; avoid implicit conversions

Lightweight Validation

- Predicates; guard clauses; clear messages

Common Pitfalls

- Float rounding; parsing localized strings

Lab Preview

- Formatters & validators

Module 14: Error Handling & Defensive JS

Fundamentals

- try/catch/finally; throw early; group failures

Custom Errors

- Extend Error; metadata; debugging value

Sync vs Async Errors

- Rejections vs throws; `.catch` vs `try/catch`

Guard Clauses/Invariants

- Validate inputs; assert impossible states

Fail Soft

- ?? and ?. for safe defaults; actionable logs

Logging Discipline

- group/table/time; reduce noise

Error-First Checklist

- Inputs, fallbacks, useful logs, recovery paths

Lab Preview

- Harden a brittle function

Module 15: Program Structure & Reuse

Public Surface

- Export minimum; document APIs; avoid cycles

Dependency Boundaries

- Pure core; impure edges; explicit data flow

Configuration as Data

- data-* and JSON

Composition > Inheritance

- Build from small utilities; thin classes

Naming & Discoverability

- Feature-first structure; clear entry points

Lightweight Testing

- Small harnesses; deterministic IO

Review Checklist

- Coupling, naming, docs, side effects

Lab Preview

- Split script into modules

Module 16: Async I — Callbacks to Promises

Event Loop

- Tasks, microtasks, rendering

Callbacks → Promises

- Error-first; inversion of control

States and Chaining

- pending/fulfilled/rejected; return vs throw

Combinators

- all, allSettled, race, any

Micro vs Macro

- `queueMicrotask`; order gotchas

AbortController

- Cancel fetch; prevent leaks/stale updates

Where Promises Fail

- Missing returns/awaits; swallowed rejections

Lab Preview

- Promise demos & combinators

Module 17: Async II — `async/await`

Syntax & Handling

- `async/await`; `try/catch`

Sequential vs Parallel

- `await` vs `Promise.all`; partial failures

Timeouts/Retries

- race with timeout; simple backoff

Streams Preview

- Progressive processing (conceptual)

Progress & State

- Spinner, disabled UI, optimistic updates

Manual Testing

- Fake delays; milestone logs

Pitfalls

- Double awaits; missing returns; exceptions vs rejections

Lab Preview

- Refactor chains; timeout; loading state

Module 18: fetch + JSON + CORS

Essentials

- Request/Response; parse JSON; errors

Status/Headers

- `response.ok`; `content-type`; caching

CORS

- Browser enforcement; preflight; public APIs

POST JSON

- Headers; JSON.stringify; server errors

Retry/Backoff

- Idempotency; jitter

Security

- Sanitize input; avoid secrets; rate limits

Progressive Enhancement

- Offline/slow cues; cached fallback

Lab Preview

- Small fetch UI with robust states

Module 19: URL & Forms APIs

URLs as State

- `URLSearchParams`; encode filters

FormData

- submit, preventDefault; progressive enhancement

Serialize Filters

- Sync inputs↔URL; debounce; restore on load

Deep Links

- Reproduce state via URL; back/forward

Accessibility

- Labels, roles, focus, keyboard-first

Patterns

- Tables/search; invalid/conflicting params

To History API

- push/replace; sync with navigation

Lab Preview

- Queryable list

Module 20: History & Storage APIs

History Essentials

- `pushState/replaceState/popstate`; minimal state

SPA-Lite

- Link interception; routing table; scroll/focus

Storage APIs

- localStorage vs sessionStorage; quotas

Privacy/Security

- Avoid sensitive data; respect prefs

Sync UI

- One source of truth; avoid drift

Persistence Patterns

- Drafts, preferences, last view; recovery

Pitfalls

- Over-storing; stale reads; versioning

Lab Preview

- SPA-lite with preferences

Module 21: Timers, Animation, Intersection Observer

Timers

- `setTimeout` vs `setInterval`; drift; cleanup

rAF

- Align with paint; throttle updates

Intersection Observer

- Lazy-load images; subtle effects

Scroll/Resize Perf

- Debounce/throttle; avoid reflows; batch DOM writes

Motion & A11y

- prefers-reduced-motion; tasteful effects

Measuring

- Performance panel basics; simple metrics

Framework Thresholds

- When to consider a framework

Lab Preview

- Lazy images + fade-in

Module 22: Organizing Small Apps

Folders & Barrels

- Feature-first vs layer-first; when barrels help

API Boundary Modules

- Mockable; separate shape vs transport

State Without Frameworks

- Minimal shared state; events/observers

DOM Update Patterns

- Templates/fragments; partial updates

Error Boundaries (Conceptual)

- Contain failures; friendly fallbacks

Build vs Ship

- What to bundle; budgets

Lab Preview

- Final project skeleton

Module 23: Choosing & Integrating a Public API

Picking the API

- Stability, CORS, auth, limits; shape/docs

UI Around Latency

- Loading/skeletons/retries; partial/delayed data

Pagination vs Infinite Scroll

- UX/perf trade-offs; accessibility

Error/Empty First

- Design retries; helpful failures

Accessible Data Views

- Roles/semantics; focus; keyboard

Small CSS System

- Tokens, utilities, theming

Finish Line Checklist

- Features, docs, perf, a11y; rubric

Lab Preview

- Fetch + render with robust states

Module 24: Polish & Persistence

Search/Filter + URL

- Sync both ways; debounce; deep link

Detail View & Routing

- History API; preserve context; focus on nav

Persistence

- Theme + last filter in localStorage; versioning

Accessibility Sweep

- Labels, headings, contrast, focus rings

Performance Pass

- Network + rendering; measure before/after

Packaging

- What to ship; cache headers; README

Optional Extras

- Clipboard, share, copy URL

Lab Preview

- Route, persist, and ship