

CSS: The Complete Guide (Flexbox, Grid & Sass)

Learn modern CSS from the ground up or deepen your existing skills. Essential for every web developer building layouts today.

Programming Instructor: Ryan Mitchell • 5 sections • 45 lessons • 15.50h total

[View this course online](#) [Browse all courses](#)

About this course

CSS seems simple until your layout breaks, you fight the cascade, or you need to build something truly responsive. This course cuts through the confusion. You'll start with the core concepts like the box model, specificity, and units, then move methodically into modern layout systems.

We'll build practical projects using Flexbox for component alignment and CSS Grid for full page layouts. The final section integrates Sass for a professional workflow, covering variables, architecture, and modern tooling. Expect clear explanations, real debugging scenarios, and code you can use immediately.

Course curriculum

CSS Fundamentals

- **How CSS Works with HTML (12m)**
Understand the relationship between HTML structure and CSS presentation, including the browser's rendering process.
- **Selectors, Properties, and Values (18m)**
Write your first rules. Learn about element, class, ID, and attribute selectors to target HTML precisely.
- **The Cascade, Specificity, and Inheritance (25m)**
Solve styling conflicts. A step-by-step guide to how the browser decides which CSS rule wins.
- **Colors, Backgrounds, and Gradients (15m)**
Apply color in various formats (HEX, RGB, HSL) and create simple background effects.
- **Units: px, em, rem, %, vw, vh (22m)**
Choose the right unit for responsive and accessible design. Compare absolute vs. relative units.
- **Basic Typography with CSS (18m)**
Control font properties, line height, and spacing to improve text readability.
- **Linking Stylesheets & CSS Placement (10m)**
Compare inline, embedded, and external stylesheets. Learn the best practice for organizing your code.

Core Layout & The Box Model

- **Mastering the Box Model (28m)**
Visualize every element as a box. Deep dive into margin, border, padding, and content with box-sizing.
- **Display Types: block, inline, inline-block (20m)**
Control how elements flow in the document. Understand the default display of common HTML tags.
- **Normal Flow vs. Out-of-Flow Positioning (15m)**
Introduction to static, relative, absolute, and fixed positioning contexts.
- **Relative and Absolute Positioning in Practice (22m)**
Build a tooltip and a badge overlay. Learn to use position: relative as a container for absolute children.
- **Fixed and Sticky Positioning (18m)**
Create a fixed header and a sidebar that sticks. Understand the viewport and scroll contexts.
- **Floats and Clearing for Legacy Layouts (16m)**
Why floats are still relevant for simple text wrapping and legacy code. How to clear them properly.
- **Overflow and Creating Scrollable Areas (12m)**
Control what happens when content is too big for its container. Build a simple scrollable box.
- **Z-index and Stacking Contexts (25m)**
Debug why one element covers another. Understand how z-index actually works beyond simple number ordering.
- **Debugging Layout Issues (15m)**
Use browser DevTools to inspect the box model, computed styles, and layout problems in real-time.

Modern Layout with Flexbox

- **Flex Container & Flex Item Basics (18m)**
Activate Flexbox and see its immediate effect on direct children. Set up your first flexible layout.
- **Main Axis vs. Cross Axis Alignment (22m)**
Visualize the two axes. Use justify-content and align-items to distribute space.
- **Controlling Line Wrapping with flex-wrap (15m)**
Allow items to move to the next row. Use the gap property for consistent spacing.
- **Growing, Shrinking, and Sizing Flex Items (28m)**
Control how items grow and shrink with flex-grow, flex-shrink, and flex-basis. The shorthand flex property.
- **Reordering Items with order (10m)**
Change visual order without altering HTML. Practical use cases for navigation and content prioritization.
- **Building a Responsive Navigation Bar (25m)**
Step-by-step project: Create a horizontal nav that stacks on mobile using Flexbox properties.
- **Creating Card Layouts with Flexbox (20m)**
Align card content consistently and make cards equal height regardless of text length.
- **Centering Everything with Flexbox (12m)**
The definitive way to center elements horizontally and vertically. Solve the classic centering problem.

Advanced Layout with CSS Grid

- **Defining Your First Grid Container (20m)**
Create a grid with explicit rows and columns. Understand the difference between grid and flexbox.
- **The fr Unit, repeat(), and auto-sizing (25m)**
Use fractional units for flexible tracks. Write cleaner code with repeat() and auto keywords.
- **minmax() and Responsive Grids without Media Queries (28m)**
Build grids that adapt using auto-fill/auto-fit with minmax(). Create a responsive image gallery.
- **Placing Items with grid-template-areas (22m)**
Design layouts visually with named areas. Create a common header/sidebar/content/footer page.
- **Line-Based Placement for Precise Control (18m)**
Place items on specific grid lines. Use the grid-column and grid-row shorthand properties.
- **Alignment in Grid: justify and align (20m)**
Control item and content alignment along both axes in a grid context.
- **Implicit Grids and Auto-placement (15m)**
Handle content that overflows your defined tracks. Use grid-auto-flow for dense packing.
- **Nested Grids and Subgrid (32m)**
Align children of grid items to the outer grid using subgrid for perfect alignment in complex cards.
- **Combining Grid and Flexbox in a Project (30m)**
Use Grid for the overall page structure and Flexbox for component-level alignment. A practical case study.
- **Debugging Grid Layouts (14m)**
Use browser DevTools Grid inspector and overlay tools to visualize tracks, lines, and areas.

Sass & Professional Workflow

- **What Sass Solves and Setup** (18m)
Why vanilla CSS isn't enough for large projects. Install Sass and compile your first file.
- **Variables and Nesting for Cleaner Code** (20m)
Store colors, fonts, and spacing in variables. Nest selectors to mirror HTML structure.
- **Partials and @use/@forward for Imports** (25m)
Organize code into multiple files. The modern way to share variables and mixins across files.
- **Creating Reusable Mixins** (28m)
Write mixins for common patterns like flexbox centering, text truncation, and responsive breakpoints.
- **Functions and Custom Logic** (22m)
Write functions to calculate values, lighten colors, or generate spacing scales.
- **@extend, Placeholders, and Silent Classes** (15m)
Share a set of properties without extra HTML classes. Understand the benefits and trade-offs.
- **Sass Architecture: The 7-1 Pattern** (30m)
Organize a large project with a scalable folder structure. Integrate the BEM methodology.
- **Custom Properties (CSS Variables) vs. Sass Variables** (20m)
When to use each. Use CSS variables for runtime theming and Sass for build-time logic.
- **Setting Up a Build Process** (25m)
Automate Sass compilation, minification, and vendor prefixing with a simple build script.
- **Introduction to PostCSS and Modern Plugins** (18m)
Use PostCSS for additional transformations. Explore plugins for future CSS syntax today.
- **Final Project: Building a Style Guide with Sass** (45m)
Apply everything learned. Create a live style guide with variables, mixins, and a component library.

Discover our blog

Learn more with free articles written by our professors and guest experts.

Latest article: **Flight Attendant Preparation: What to Study Before You Apply**

[Visit the blog](#) [All courses](#) [This course](#)