
WEB ENGINEERING FUNDAMENTALS

CSS Custom Properties

A Complete Beginner's Guide to Writing Maintainable CSS

Learn how CSS Custom Properties (commonly called CSS Variables) help you write cleaner, reusable, and scalable styles. Understand how professional developers manage colors, spacing, typography, themes, and design systems.

haas.dev • dev-roast-app.vercel.app

Table of Contents

1. Introduction
2. What Are Custom Properties?
3. Why CSS Variables Matter
4. How CSS Variables Work
5. Defining Variables
6. Using Variables
7. Variable Scope
8. Global vs Local Variables
9. Fallback Values
10. Building Themes with Variables
11. Variables in Design Systems
12. Real-World Examples
13. Common Beginner Mistakes
14. Practical Action Plan
15. Mini Project
16. Key Takeaways
17. Summary Page
18. CSS Variables Cheat Sheet
19. Related Resources
20. Recommended Next Learning Path

1. Introduction

Imagine you are building a website with 50 pages.

The primary brand color appears in:

- buttons
- headings
- links
- cards
- icons
- navigation
- footer

A few months later, the client decides to change the primary color.

Without CSS Variables, you might need to search through hundreds of CSS rules and manually update every occurrence.

This approach is slow, error-prone, and difficult to maintain.

CSS Custom Properties solve this problem by allowing you to define a value once and reuse it throughout your stylesheet.

Instead of changing the value in many places, you update it in one location.

2. What Are CSS Custom Properties?

CSS Custom Properties are reusable values stored inside variables.

These values can represent:

- colors
- font sizes
- spacing
- border radius
- shadows
- animation durations
- layout dimensions

A variable acts like a named container.

Instead of repeating the same value throughout your CSS, you reference the variable wherever it is needed.

3. Why CSS Variables Matter

Modern websites often contain thousands of lines of CSS.

Repeating values throughout a project creates several problems:

- inconsistent styling
- difficult maintenance
- higher risk of mistakes
- slower updates

CSS Variables improve:

- consistency
- readability
- maintainability
- scalability

They are also a key building block of modern design systems.

4. How CSS Variables Work

The process has two steps.

Step 1

Define a variable with a meaningful name.

Examples include:

- primary color
- spacing
- font size
- border radius

Step 2

Reference that variable wherever the value is needed.

The browser replaces the variable with its stored value when rendering the page.

This allows one definition to control many elements.

5. Defining Variables

Variables are commonly defined in the `:root` selector.

This makes them available throughout the entire document.

Typical variables include:

- brand colors
- typography scale
- spacing scale
- shadows
- transition durations

Keeping global variables together creates a single source of truth for your design.

6. Using Variables

Once defined, variables can be used in almost every CSS property.

Examples include:

- text color
- background color
- margins
- padding
- borders
- widths
- heights

Instead of repeating literal values, you reference the appropriate variable.

This makes future updates much easier.

7. Variable Scope

Not every variable needs to be global.

Some values are only relevant to a specific component.

For example:

- a card component
- a navigation bar
- a modal dialog

Local variables help keep component styles isolated and easier to maintain.

8. Global vs Local Variables

Global Variables

Available everywhere.

Best for:

- colors
- spacing scales
- typography
- border radius

Local Variables

Available only within a specific element or component.

Useful for:

- component-specific sizes
- unique animations
- temporary overrides

Professional developers often combine both approaches.

9. Fallback Values

Sometimes a variable may not exist.

CSS allows developers to provide a fallback value.

If the variable cannot be found, the browser uses the fallback instead.

Fallbacks improve resilience and make components easier to reuse.

10. Building Themes with Variables

One of the biggest advantages of CSS Variables is theme switching.

Instead of rewriting hundreds of CSS rules, developers change a small set of variables.

Examples include:

- Light theme
- Dark theme
- Seasonal color schemes
- Brand customization

Modern websites often use CSS Variables as the foundation for theme management.

11. CSS Variables in Design Systems

Large organizations maintain consistent user interfaces through design systems.

CSS Variables allow teams to define reusable design tokens such as:

- primary colors
- secondary colors
- spacing values
- font sizes
- shadows
- border radius

When a design token changes, every component using that token updates automatically.

This reduces maintenance and keeps the interface visually consistent.

Learn More

Read [Responsive CSS Units](#) to understand how variables can store reusable spacing and typography values.

12. Real-World Examples

E-commerce Website

The primary button color is stored as a global variable.

Changing one variable updates every product page.

News Website

Typography sizes are managed with variables, ensuring consistent headings and paragraph styles across all articles.

haas.dev

Colors, spacing, typography, border radius, and shadows can all be managed through CSS Variables.

If the brand evolves, updating a few variables refreshes the appearance of the entire platform.

13. Common Beginner Mistakes

- Giving variables unclear names.
- Creating too many unnecessary variables.
- Mixing hardcoded values with variables inconsistently.
- Using global variables for component-specific values.
- Forgetting that variables should improve maintainability, not add complexity.

14. Practical Action Plan

Create a small style guide for a project.

Define variables for:

- primary color
- secondary color
- background color
- heading sizes
- body text size
- spacing
- border radius

Apply these variables consistently throughout the project.

Then change the primary color and observe how quickly the design updates.

15. Mini Project

Build a simple landing page using CSS Variables for:

- colors
- typography
- spacing
- buttons
- cards

Add a dark mode by changing only the variable values.

Notice how little CSS needs to change.

16. Key Takeaways

- CSS Custom Properties store reusable values.
- Variables improve consistency and maintainability.
- Global variables define shared design tokens.
- Local variables support component customization.
- Variables make theme switching simple.
- Modern design systems rely heavily on CSS Variables.

17. Summary Page

CSS Variables Cheat Sheet

Store reusable values.

Prefer meaningful variable names.

Use global variables for shared styles.

Use local variables for component-specific styles.

Build themes by updating variables.

Reduce repeated values.

Improve maintainability.

18. CSS Variables Cheat Sheet

A quick-reference checklist you can keep beside your editor while you build.

Define shared design tokens in `:root`

Give variables clear, meaningful names

Use global variables for colors, spacing, and typography scales

Use local variables for component-specific values

Provide fallback values for resilience

Build themes (light/dark) by swapping variable values

Avoid mixing hardcoded values with variables inconsistently

19. Related Resources

[CSS Fundamentals](#)

Why read it: Strengthen your understanding of core CSS concepts.

[Responsive CSS Units](#)

Why read it: Combine variables with scalable sizing.

[Responsive Web Design](#)

Why read it: Use variables to maintain consistency across responsive layouts.

[CSS Grid](#)

Why read it: Apply variables to spacing and layout values.

[CSS Flexbox](#)

Why read it: Build reusable component layouts using design tokens.

20. Recommended Next Learning Path

Step 1

Responsive Web Design

↓

Step 2

CSS Media Queries

↓

Step 3

Responsive CSS Units

↓

Step 4

Mobile-First Design

↓

Step 5

Responsive Images & Modern Image Optimization

↓

Step 6

CSS Custom Properties (CSS Variables) (Current PDF)

↓

Step 7

CSS Transforms

↓

Step 8

CSS Transitions

↓

Step 9

CSS Animations

↓

Step 10

Build Your First Complete Responsive Website