

How can CSS style a web page?

CSS stands for **Cascading Style Sheets**. It controls the look of your HTML: colors, fonts, spacing, borders and layout.

This page is for **absolute beginners** who already know a little HTML. First you build the structure with HTML, then you use CSS to make the page look nicer.

1. SEPARATING CONTENT AND DESIGN

HTML for structure, CSS for style

HTML describes what the content is. CSS describes how the content should look. This makes your pages easier to change and reuse.

```
<!-- HTML -->
<h1 class="title">Welcome to my page</h1>
<p>This is a paragraph of text.</p>
CSS style
<style>
.title{
  color: blue;
  font-family: Arial, sans-serif;
}
</style>
```

The same HTML can look very different when you change the CSS rules.

In this example the HTML and CSS are shown together, but in a real project the CSS is often kept in a separate file.

2. ORGANIZING CSS

Use reusable rules

For a quick Sandbox experiment, CSS can go inside a `<style>` block. For a website with several pages, keep reusable rules in a separate stylesheet:

```
<head>
  <link rel="stylesheet" href="styles.css">
</head>
/* styles.css */
.highlight {
  background-color: #1e3a8a;
  color: white;
  padding: 0.2rem 0.35rem;
}
```

Use classes for styles that can be reused. Avoid placing presentation rules directly in an element's `style` attribute as the normal approach, because inline styles are harder to reuse and maintain.

3. RESPONSIVE AND ACCESSIBLE CSS

Make the design usable for everyone

Use relative units such as [rem](#), keep foreground and background colors readable, and preserve a clear keyboard focus indicator.

```
p {
  font-size: 1.125rem;
  line-height: 1.6;
}
button:focus-visible {
  outline: 3px solid #f59e0b;
  outline-offset: 3px;
}
img {
  max-width: 100%;
  height: auto;
}
```

Use CSS animation instead of obsolete scrolling elements, and disable nonessential motion when the visitor requests reduced motion.

```
.welcome { animation: enter 600ms ease-out both; }
@keyframes enter {
  from { opacity: 0; transform: translateY(0.75rem); }
  to { opacity: 1; transform: translateY(0); }
}
@media (prefers-reduced-motion: reduce) {
  .welcome { animation: none; }
}
```

4. PRACTICE

What can you try yourself?

- Change the background color of the page.
- Set paragraph text with [rem](#) and test browser zoom at 200%.
- Add visible [:focus-visible](#) styling to a button.

Copy a simple HTML page into the Sandbox, then add a `<style>` block or link a CSS file and see how the layout changes.

5. WHAT TO DO NEXT?

Continue your CSS and JavaScript learning path

After you understand the basic idea of CSS, continue with these steps:

- **Practise more CSS:** go to the [CSS codes page](#) and try the ready-made CSS examples in the Sandbox.
- **Combine HTML and CSS:** start with a simple HTML page and slowly add more CSS rules to see how each rule changes the page.
- **Get ready for interactivity:** read the [“How can JavaScript make a web page interactive?”](#) explanation to see what JavaScript can add on top of HTML and CSS.
- **Practise with assignments:** open the [HTML, CSS and JavaScript exercises \(PDF\)](#) and try to style your own small web pages.

Take it step by step: first learn HTML structure, then use CSS for colors and layout, and finally use JavaScript to make your pages interactive.