

Web Coding Camp — Day 2

CSS: Color, Type & the Box Model

David Pouliot
Southern Oregon University

Today

Yesterday you got a *taste* of CSS. Today you go deeper and make the page truly *yours*.

- Moving CSS into its own file
- Selectors — tags, **classes**, and **ids**
- Colours, backgrounds, and fonts
- The box model — margin, border, padding
- A bit more CSS — hover, gradients, variables

Goal: take yesterday's plain page and give it a colour palette, a nice font, and a "card" look.

What Is CSS?

Give CSS its own file

Yesterday you used a `<style>` block. Today, move your styles into a separate file and `<link>` it — cleaner, and it can style every page.

```
1 <head>
2   <title>My Page</title>
3   <link rel="stylesheet" href="style.css">
4 </head>
```

Now make a file called `style.css` next to your HTML. Everything today goes in there.

The shape of a CSS rule

Pick something, then list what to change.

```
1 h1 {  
2   color: purple;  
3   text-align: center;  
4 }
```

- h1 — the **selector** (what to style)
- color — the **property**
- purple — the **value**
- Don't forget the ; after each line and the { }.

Selectors

Three ways to pick elements

```
1 p          { color: gray; }      /* every <p> */
2 .note      { color: red; }       /* every class="note" */
3 #header    { color: blue; }      /* the one id="header" */
```

- By **tag** — styles *all* of that tag.
- By **class** (.) — reusable; put it on many elements.
- By **id** (#) — unique; one element per page.

The <div>: a box to group things

A <div> is a plain **container**. On its own it does nothing — but it lets you group elements and style them as one block.

```
1 <div class="card">
2   <h2>Dog Walking Service</h2>
3   <p>Reliable walks in your neighborhood.</p>
4 </div>
```

- Wrap related elements in a <div>, give it a class, then style the whole box at once.
- You'll use <div> constantly — for cards, sections, and layout.

Classes in the HTML

Add `class="..."` in your HTML, then style that class in CSS.

```
1 <p class="note">Watch out for this!</p>
```

```
1 .note {  
2   color: darkred;  
3   font-weight: bold;  
4 }
```

Classes are how you style *some* paragraphs but not all of them.

Ids: for one special element

An id names *one* element: `id="..."` in the HTML, `#` in the CSS.

```
1 <div id="hero">Welcome!</div>
```

```
1 #hero {  
2     background: #1a1a2e;  
3     color: white;  
4 }
```

Class vs id: a class can repeat on many elements; an id should appear *once* per page.

Colour & Type

Colours and backgrounds

```
1 body {  
2     background: #f4f4f8;    /* a hex colour */  
3     color: #222;           /* text colour */  
4 }  
5 h1 {  
6     color: rebeccapurple;  /* named colour */  
7 }
```

- background is behind the element; color is the text.

Pick colours: htmlcolorcodes.com/color-picker gives a #hex; colors.co builds a whole palette.

Fonts and text

```
1 body {  
2   font-family: system-ui, sans-serif;  
3   line-height: 1.6;  
4 }  
5 h1 {  
6   font-size: 2rem;  
7   text-align: center;  
8 }
```

- font-family — the typeface.
- font-size — rem scales nicely; 2rem is twice the normal size.
- line-height — space between lines; 1.6 is easy to read.

A fancier font in two lines

Grab a font from `fonts.google.com`: link it, then name it.

```
1 <link href="https://fonts.googleapis.com/css2?family=Poppins&display=s  
  ↪ wap" rel="stylesheet">
```

```
1 body { font-family: "Poppins", sans-serif; }
```

The `<link>` goes in your `<head>`; the `font-family` goes in your CSS. Google Fonts gives you both to copy.

The Box Model

Every element is a box

Around every element's content are three layers:

- **padding** — space *inside* the box, around the content
- **border** — the line around the box
- **margin** — space *outside* the box, pushing others away

margin → border → padding → **content**

Padding pushes the *border* away from the text. Margin pushes *other elements* away. That's the whole difference.

Make a "card"

Padding, a background, rounded corners — the classic card look.

```
1 .card {  
2   background: white;  
3   padding: 1.5rem;  
4   border-radius: 12px;  
5   margin-bottom: 1rem;  
6   box-shadow: 0 2px 8px rgba(0,0,0,0.1);  
7 }
```

Wrap a section of HTML in `<div class="card"> ... </div>` to use it.

Center the whole page

A common trick: cap the width and auto-margin it to the middle.

```
1 body {  
2   max-width: 700px;  
3   margin: 0 auto;    /* 0 top/bottom, auto left/right = centered */  
4   padding: 2rem;  
5 }
```

margin: 0 auto is the "center this block" spell. You'll use it constantly.

A Bit More CSS

Hover effects

Add `:hover` to a selector to change something when the mouse is over it.

```
1 a {  
2   color: #6a1b9a;  
3   transition: color 0.2s;    /* animate the change */  
4 }  
5 a:hover {  
6   color: tomato;  
7 }
```

`transition` makes the change fade smoothly instead of snapping.

Three finishing touches

```
1  /* a gradient background */
2  body { background: linear-gradient(#a18cd1, #fbc2eb); }
3
4  /* a soft shadow under a card */
5  .card { box-shadow: 0 4px 12px rgba(0,0,0,0.2); }
6
7  /* name a colour once, reuse it everywhere */
8  :root { --brand: #6a1b9a; }
9  h1 { color: var(--brand); }
```

A --variable lets you change your theme colour in *one* place.

Project: Style Your Page

Restyle yesterday's page

Link a stylesheet and give your page a look. Starter to build on:

```
1 body {  
2   font-family: system-ui, sans-serif;  
3   background: #f4f4f8;  
4   color: #222;  
5   max-width: 700px;  
6   margin: 0 auto;  
7   padding: 2rem;  
8 }  
9 h1 { color: #6a1b9a; text-align: center; }  
10 img { border-radius: 12px; width: 200px; }
```

Make it yours + stretch goals

Your turn: pick a palette (coolors.co), a Google Font, and spacing you like. Give your main section the card look.

Stretch goals:

- A hover effect on links (`a:hover`)
- A CSS gradient background
- A box-shadow on an image or card
- Style a `<button>` even if it does nothing yet

Full styled example is on the camp website if you get stuck.