**1.** CSS Reset / Normalize

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Simple CSS Reset Page</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: sans-serif;

background-color: #f4f4f4;

padding: 20px;

}

h1 {

background-color: #333;

color: #fff;

padding: 10px;

}

p {

background-color: #fff;

padding: 10px;

margin-top: 10px;

}

</style>

</head>

<body>

<h1>CSS Universal Reset</h1>

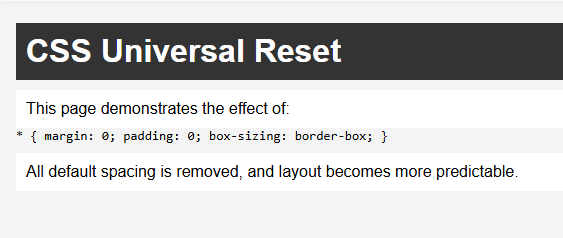
<p>This page demonstrates the effect of:</p>

<pre>\* { margin: 0; padding: 0; box-sizing: border-box; }</pre>

<p>All default spacing is removed, and layout becomes more predictable.</p>

</body>

</html>

**Output:** 

 A styled page with no unexpected spacing or layout glitches

 Clean use of padding and margins that you control explicitly

2. **Centering Content**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<title>Flexbox Centering</title>

<style>

/\* Reset and basic styling \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body, html {

height: 100%;

}

.container {

display: flex;

justify-content: center;

align-items: center;

height: 100vh; /\* Full viewport height \*/

background-color: #f0f0f0;

}

.box {

background-color: #4CAF50;

color: white;

padding: 40px 60px;

font-size: 1.5rem;

border-radius: 8px;

}

</style>

</head>

<body>

<div class="container">

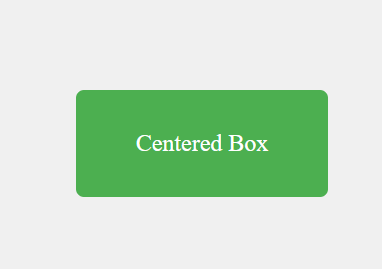
<div class="box">Centered Box</div>

</div>

</body>

</html>

o/p:



 A **green background** that fills the whole screen

 A white **green box** with white text centered in the middle

3. Responsive Units

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Responsive Container Example</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: sans-serif;

background-color: #f9f9f9;

display: flex;

justify-content: center;

padding: 2rem;

}

.container {

width: 100%;

max-width: 1200px;

padding: 1rem;

font-size: clamp(1rem, 2vw, 1.5rem);

background-color: #ffffff;

border: 1px solid #ccc;

box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);

}

</style>

</head>

<body>

<div class="container">

<h1>Responsive Container</h1>

<p>

This container uses <code>width: 100%</code>, <code>max-width: 1200px</code>, and

<code>font-size: clamp(1rem, 2vw, 1.5rem)</code> to stay readable and responsive on any screen size.

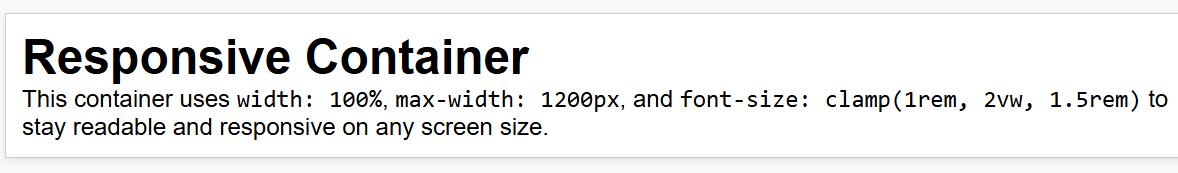
</p>

</div>

</body>

</html>

o/p



 A white, centered content box

 Automatically adapts to screen size

 Font scales smoothly between 1rem and 1.5rem depending on the viewport width

4. Hover Effects

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Button Hover Effect</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: sans-serif;

background-color: #f4f4f4;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

button {

padding: 12px 24px;

font-size: 1rem;

border: none;

border-radius: 5px;

background-color: #eee;

color: #333;

cursor: pointer;

transition: all 0.3s ease;

}

button:hover {

background-color: #333;

color: white;

transform: scale(1.05);

}

</style>

</head>

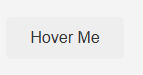
<body>

<button>Hover Me</button>

</body>

</html>

o/p

 Highlight the background color with black.

5. Transitions

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>CSS Transition Example</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-color: #f5f5f5;

font-family: Arial, sans-serif;

}

.box {

width: 150px;

height: 150px;

background-color: #4CAF50;

border-radius: 8px;

transition: all 0.3s ease;

}

.box:hover {

background-color: #333;

transform: scale(1.1);

border-radius: 50%;

}

</style>

</head>

<body>

<div class="box"></div>

</body>

</html>

o/p

transition for above one to circle with black color.

6. Box Shadows and Borders

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Box Shadow and Border Radius</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-color: #f2f2f2;

font-family: sans-serif;

}

.card {

width: 300px;

padding: 20px;

background-color: white;

border-radius: 8px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

text-align: center;

}

h2 {

margin-bottom: 10px;

}

p {

color: #555;

}

</style>

</head>

<body>

<div class="card">

<h2>Card Title</h2>

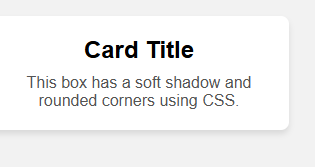
<p>This box has a soft shadow and rounded corners using CSS.</p>

</div>

</body></html>

o/p

----



* A centered **white card** with:
  + Rounded corners (border-radius: 8px)
  + A subtle drop shadow (box-shadow)
* Clean and modern look — ideal for UI components like cards, modals, or alerts

7. Sticky Navigation

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Sticky Nav Example</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: Arial, sans-serif;

line-height: 1.6;

}

nav {

position: sticky;

top: 0;

z-index: 1000;

background: white;

padding: 1rem;

border-bottom: 1px solid #ccc;

}

nav ul {

display: flex;

gap: 20px;

list-style: none;

}

nav a {

text-decoration: none;

color: #333;

font-weight: bold;

}

main {

padding: 2rem;

}

section {

height: 100vh; /\* Each section takes up full screen height \*/

padding: 1rem;

border-bottom: 1px solid #ddd;

}

h2 {

margin-bottom: 1rem;

}

</style>

</head>

<body>

<nav>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">About</a></li>

<li><a href="#">Services</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

<main>

<section>

<h2>Section 1</h2>

<p>Scroll down to see the sticky nav stay at the top.</p>

</section>

<section>

<h2>Section 2</h2>

<p>More content below to test the sticky behavior.</p>

</section>

<section>

<h2>Section 3</h2>

<p>The nav remains visible even as you scroll.</p>

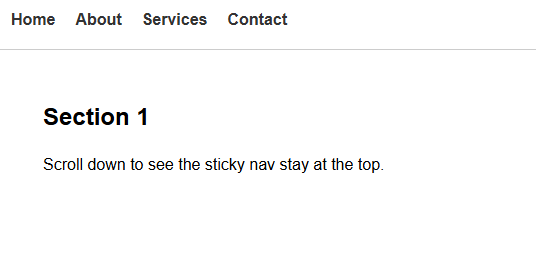
</section>

</main>

</body>

</html>

o/p

Its just like afreeze header.

8. Media Queries for Mobile

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Responsive Menu</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: Arial, sans-serif;

padding: 20px;

}

nav {

background-color: #333;

padding: 1rem;

}

.menu {

display: flex;

justify-content: space-around;

list-style: none;

}

.menu li a {

color: white;

text-decoration: none;

padding: 0.5rem 1rem;

display: block;

}

.menu li a:hover {

background-color: #444;

}

/\* Responsive layout: stack menu items vertically \*/

@media (max-width: 768px) {

.menu {

flex-direction: column;

align-items: center;

}

}

</style>

</head>

<body>

<nav>

<ul class="menu">

<li><a href="#">Home</a></li>

<li><a href="#">About</a></li>

<li><a href="#">Services</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

<main>

<p>Resize the browser window below 768px to see the menu stack vertically.</p>

</main>

</body>

</html>

o/p

highlight the square color for column wise.

9. Variables for Themes

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>CSS Variables Example</title>

<!-- Google Font (Roboto) -->

<link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">

<style>

:root {

--primary-color: #3498db;

--font-main: 'Roboto', sans-serif;

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: var(--font-main);

background-color: #f4f4f4;

padding: 2rem;

}

.card {

background-color: white;

padding: 2rem;

border-radius: 8px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

max-width: 400px;

margin: auto;

}

h1 {

color: var(--primary-color);

margin-bottom: 1rem;

}

p {

color: #555;

}

</style>

</head>

<body>

<div class="card">

<h1>CSS Variables</h1>

<p>

This page uses custom properties defined in <code>:root</code> for the primary color and font.

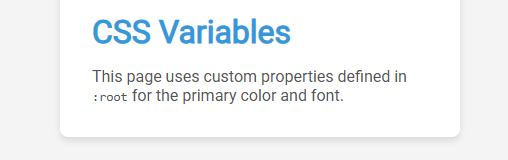
</p>

</div>

</body>

</html>

o/p



 A centered card with a Roboto font and a blue heading

 The color and font are both pulled from the variables declared in :root

10. Z-Index for Layering

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Simple Modal Example</title>

<style>

body {

font-family: Arial, sans-serif;

padding: 2rem;

}

.modal {

position: fixed;

top: 0;

left: 0;

width: 100vw;

height: 100vh;

background: rgba(0, 0, 0, 0.5);

z-index: 9999;

display: flex;

justify-content: center;

align-items: center;

}

.modal-content {

background: white;

padding: 2rem;

border-radius: 8px;

text-align: center;

max-width: 400px;

}

.close-btn {

margin-top: 1rem;

padding: 0.5rem 1rem;

background-color: #3498db;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

</style>

</head>

<body>

<h1>Page Content</h1>

<p>This is some background content behind the modal.</p>

<!-- Modal -->

<div class="modal" id="modal">

<div class="modal-content">

<h2>Modal Window</h2>

<p>This modal is fixed and has a high z-index.</p>

<button class="close-btn" onclick="document.getElementById('modal').style.display='none'">Close</button>

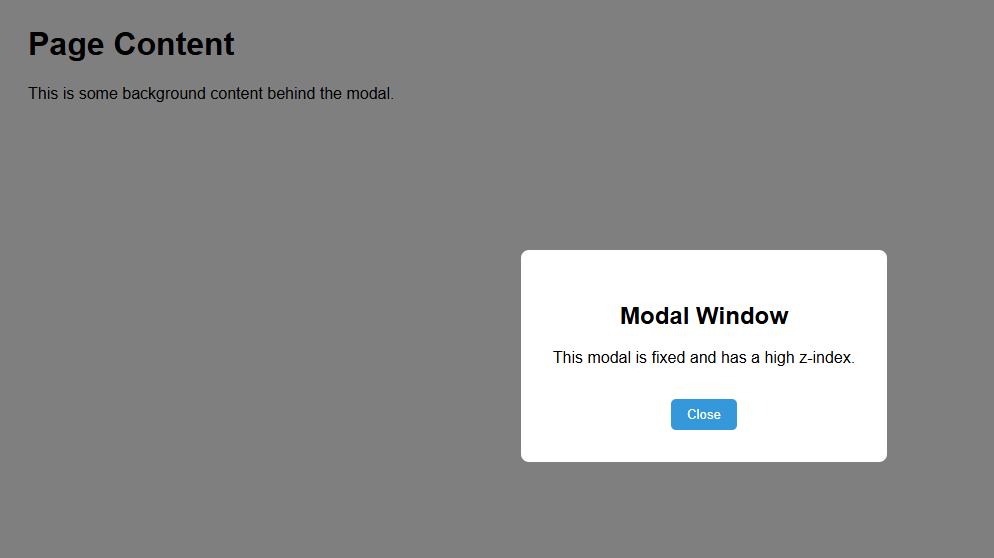
</div>

</div>

</body>

</html>

o/p



 A dark overlay modal covers the entire screen.

 A white box in the center contains a message and a **"Close"** button.

 Clicking "Close" will hide the modal by setting display: none.