

Experiment 3.1

Code:

```
<html>
<head>
<title>Pricing Plans</title>
<style>
h1{text-align:center;font-family:sans-serif;}
.pricing-container{display:flex;flex-wrap:wrap;}
.pricing-card{display:flex;flex-direction:column;justify-content:space-between;flex:0 0 200px;border:2px solid black;padding:20px;text-align:center;min-height:150px;}
.basic-plan{order:0;}
.pro-plan{order:1;flex-grow:2;}
.premium-plan{order:2;}
h2,p{margin:10px 0;font-family:sans-serif;}
</style>
</head>
<body>
<h1>Pricing Plans</h1>
<div class="pricing-container">
<div class="pricing-card basic-plan">
<h2>Basic</h2>
<p>$9/month</p>
</div>
<div class="pricing-card pro-plan">
<h2>Pro</h2>
<p>$19/month</p>
</div>
<div class="pricing-card premium-plan">
<h2>Premium</h2>
<p>$29/month</p>
</div>
</div>
</body>
</html>
```

Output:

Pricing Plans

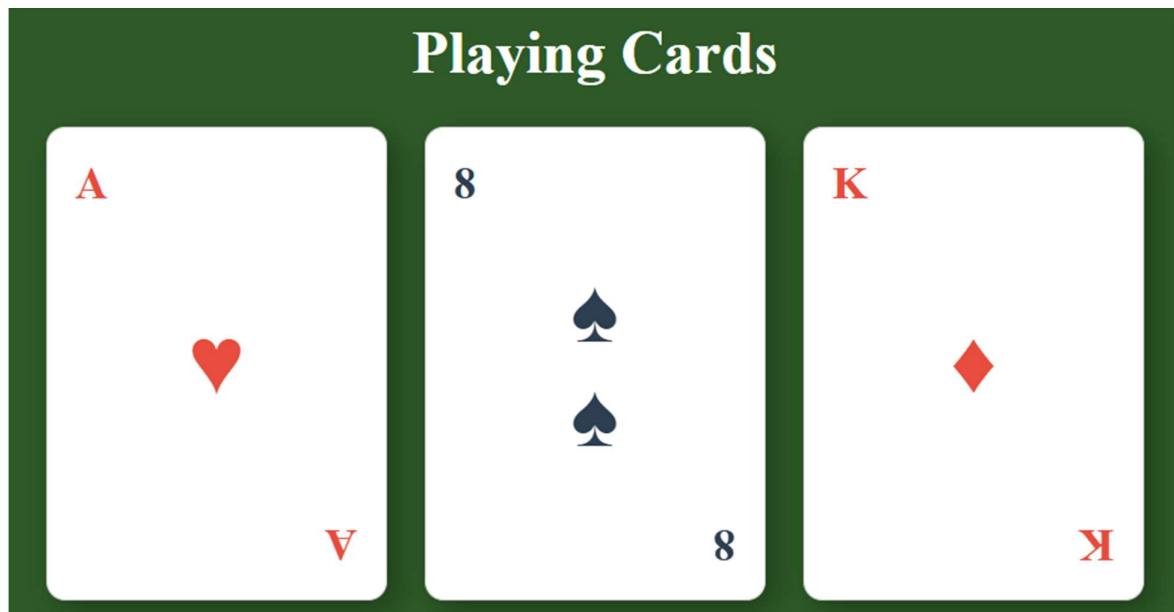
Basic	Pro	Premium
\$9/month	\$19/month	\$29/month

Experiment 3.2

Code:

```
<html>
<head>
<title>Playing Cards</title>
<style>
body{background-color:#2d5a27;margin:0;padding:50px;font-family:serif;}
#playing-cards{display:flex;justify-content:center;flex-wrap:wrap;gap:20px;}
.card{display:flex;justify-content:space-between;width:150px;height:220px;background:white;border-radius:10px;padding:15px;box-shadow:5px 5px 15px rgba(0,0,0,0.3);border:1px solid #ccc;}
.left{align-self:flex-start;font-size:1.5rem;font-weight:bold;}
.middle{align-self:center;display:flex;flex-direction:column;font-size:3rem;}
.right{align-self:flex-end;font-size:1.5rem;font-weight:bold;transform:rotate(180deg);}
.red{color:#e74c3c;}
.black{color:#2c3e50;}
</style>
</head>
<body>
<h1 align="center" style="color:#ffffff;">Playing Cards</h1>
<main id="playing-cards">
<div class="card red">
<div class="left">A</div>
<div class="middle">
<div>&hearts;</div>
</div>
<div class="right">A</div>
</div>
<div class="card black">
<div class="left">8</div>
<div class="middle">
<div>&spades;</div>
<div>&spades;</div>
</div>
<div class="right">8</div>
</div>
<div class="card red">
<div class="left">K</div>
<div class="middle">
<div>&diams;</div>
</div>
<div class="right">K</div>
</div>
</main>
</body>
</html>
```

Output:



Experiment 3.3

Code:

```
<html><head><title>Flexbox Documentation</title>
<style>
html{scroll-behavior:smooth;}
body{font-family:'Segoe UI',sans-serif;line-height:1.5;color:#333;margin:0;display:flex;background:#fdfdfd;}
#navbar{position:fixed;width:300px;height:100%;border-right:2px solid #ddd;background:#f8f9fa;overflow-y:auto;}
#navbar header{font-size:1.5rem;font-weight:bold;padding:20px;text-align:center;border-bottom:1px solid #ddd;color:#2c3e50;}
.nav-link{display:block;padding:15px 25px;text-decoration:none;color:#444;border-bottom:1px solid #ddd;transition:background 0.3s;}
.nav-link:hover{background:#e9ecef;color:#000;}
#main-doc{margin-left:30px;padding:20px;max-width:800px;}
.main-section{margin-bottom:40px;}
.main-section header{font-size:1.8rem;font-weight:600;color:#2980b9;margin-bottom:15px;border-bottom:2px solid #eee;}
code{background:#f4f4f4;padding:2px 6px;border-radius:4px;font-family:monospace;color:#d63384;font-weight:bold;display:block;margin:10px 0; width:fit-content;}
ul{margin-bottom:10px;}
@media (max-width:768px){
body{flex-direction:column;}
#navbar{position:relative;width:100%;height:auto; border-right:none; border-bottom:2px solid #ddd;}
#main-doc{margin-left:0;padding:15px;}}
</style></head>
<body><nav id="navbar"><header>Flexbox Guide</header><a class="nav-link" href="#Introduction">Introduction</a><a class="nav-link" href="#Flex_Container">Flex Container</a><a class="nav-link" href="#Flex_Direction">Flex Direction</a><a class="nav-link" href="#Justify_Content">Justify Content</a>
<a class="nav-link" href="#Align_Items">Align Items</a></nav><main id="main-doc">
<section class="main-section" id="Introduction"><header>Introduction</header><p>Flexbox is a one-dimensional layout method for arranging items in rows or columns.</p><p>It was designed to provide a more efficient way to lay out, align, and distribute space among items in a container.</p><code>display:flex;</code><ul><li>Responsive by nature</li><li>Easier alignment</li></ul></section>
<section class="main-section" id="Flex_Container"><header>Flex Container</header><p>To start using Flexbox, you must first define a flex container.</p><p>This is done by setting the display property of an element.</p><code>.container { display: flex; }</code></section>
<section class="main-section" id="Flex_Direction"><header>Flex Direction</header><p>The flex-direction property defines which direction the container wants to stack the items.</p><p>Default value is row, but it can also be column.</p><code>flex-direction: column;</code><ul><li>row</li><li>row-reverse</li><li>column</li><li>column-reverse</li></ul></section>
<section class="main-section" id="Justify_Content"><header>Justify Content</header><p>The justify-content property is used to align the flex items along the main axis.</p><p>It helps distribute extra free space left over when flex items are either inflexible or flexible but have reached their maximum size.</p><code>justify-content: center;</code><ul><li>flex-start</li><li>flex-end</li><li>space-between</li><li>space-around</li></ul></section>
<section class="main-section" id="Align_Items"><header>Align Items</header><p>This defines the default behavior for how flex items are laid out along the cross axis.</p><p>Think of it as the justify-content version for the cross-axis.</p><code>align-items: stretch;</code></section>
</main></body></html>
```

Output:

Flexbox Guide	Introduction
Introduction	<p>Flexbox is a one-dimensional layout method for arranging items in rows or columns.</p> <p>It was designed to provide a more efficient way to lay out, align, and distribute space among items in a container.</p> <p><code>display: flex;</code></p> <ul style="list-style-type: none">• Responsive by nature• Easier alignment
Flex Container	<p>To start using Flexbox, you must first define a flex container.</p> <p>This is done by setting the display property of an element.</p> <p><code>.container { display: flex; }</code></p>
Flex Direction	<p>The flex-direction property defines which direction the container wants to stack the items.</p> <p>Default value is row, but it can also be column.</p> <p><code>flex-direction: column;</code></p> <ul style="list-style-type: none">• row• row-reverse• column
Justify Content	<p>The justify-content property is used to align the flex items along the main axis.</p> <p>It helps distribute extra free space left over when flex items are either inflexible or flexible but have reached their maximum size.</p> <p><code>justify-content: center;</code></p> <ul style="list-style-type: none">• flex-start• flex-end• space-between
Align Items	<p>This defines the default behavior for how flex items are laid out along the cross axis.</p> <p>Think of it as the justify-content version for the cross-axis.</p> <p><code>align-items: stretch;</code></p>

Experiment 3.4

Code:

```
<html><head><title>The Daily Grid</title><style>
body{font-family:'Times New Roman',serif;background-color:#f4f1ea;color:#333;margin:0;padding:20px;}
.newspaper-layout{display:grid;grid-template-columns:repeat(3,1fr);grid-template-rows:auto auto 1fr 1fr
auto;grid-template-areas:"ticker ticker ticker" "title title title" "feature-article feature-article cover-image"
"secondary-article secondary-article cover-image" "small-article1 small-article2 small-article3" "footer footer
footer";gap:20px;max-width:1200px;margin:0 auto;border:1px solid #000;padding:20px;background:white;}
.ticker-wrap{grid-area:ticker;background:#000;color:#fff;overflow:hidden;white-space:nowrap;padding:5px 0;font-
family:sans-serif;font-weight:bold;text-transform:uppercase;}
.ticker-move{display:inline-block;animation:ticker-slide 20s linear infinite;padding-left:100%;}
@keyframes ticker-slide{0%{transform:translateX(0)}100%{transform:translateX(-100%)}}
.title{grid-area:title;text-align:center;border-bottom:4px double #000;margin-bottom:10px;}
.title h1{font-size:4rem;margin:10px 0;text-transform:uppercase;letter-spacing:-2px;}
.feature-article{grid-area:feature-article;border-right:1px solid #ddd;padding-right:15px;}
.secondary-article{grid-area:secondary-article;border-right:1px solid #ddd;padding-right:15px;border-top:1px
solid #ddd;padding-top:15px;}
.cover-image{grid-area:cover-image;margin:0;display:flex;align-items:center;justify-content:center;}
.cover-image img{max-width:100%;height:auto;filter:grayscale(100%) contrast(1.2);border:1px solid #000;}
.small-article1{grid-area:small-article1;border-top:1px solid #ddd;padding-top:15px;}
.small-article2{grid-area:small-article2;border-top:1px solid #ddd;padding-top:15px;border-left:1px solid
#ddd;padding-left:15px;border-right:1px solid #ddd;padding-right:15px;}
.small-article3{grid-area:small-article3;border-top:1px solid #ddd;padding-top:15px;}
.footer{grid-area:footer;border-top:4px double #000;padding-top:20px;margin-top:20px;text-align:center;}
.subscription-form{display:flex;justify-content:center;gap:10px;margin-top:10px;}
.subscription-form input{padding:10px;border:1px solid #000;font-family:serif;width:250px;}
.subscription-form button{background:#000;color:#fff;border:none;padding:10px 20px;cursor:pointer;font-
family:sans-serif;font-weight:bold;text-transform:uppercase;}
.subscription-form button:hover{background:#333;}
h2{font-size:1.8rem;margin-top:0;}
h3{font-size:1.2rem;margin-top:0;text-transform:uppercase;}
p{line-height:1.4;text-align:justify;margin-bottom:0;}
</style></head>
<body><main class="newspaper-layout"><div class="ticker-wrap"><div class="ticker-move">Breaking News: CSS Grid
makes layouts 100% easier... New properties discovered in the browser... Stay tuned for more updates!</div></div>
<header class="title"><h1>The Daily Grid</h1></header><article class="feature-article"><h2>Breakthrough in Grid
Layouts</h2><p>Modern developers are switching to CSS Grid for complex newspaper designs. The ability to span
multiple columns and rows effortlessly has changed the landscape of web typography and design forever.</p>
</article><article class="secondary-article"><h2>Web Standards Update</h2><p>Semantic HTML continues to be the
backbone of accessible web design. Using tags like article and section ensures that screen readers can navigate
complex layouts like this one with ease.</p></article><figure class="cover-image"></figure><article class="small-article1">
<h3>Weather</h3><p>Expect clear skies and perfect rendering across all modern browsers this weekend.</p>
</article><article class="small-article2"><h3>Finance</h3><p>Flexbox stocks are steady while Grid futures are
rising rapidly in the tech sector.</p></article><article class="small-article3"><h3>Politics</h3><p>The debate
between internal and external CSS continues in the developer community.</p></article><footer class="footer"><h3>
Subscribe to our Newsletter</h3><form class="subscription-form" onsubmit="return false;"><input type="email"
placeholder="Enter your email" required><button type="submit">Join Now</button></form></footer></main></body>
</html>
```

Output:

BREAKING NEWS: CSS GRID MAKES LAYOUTS 100% EASIER... NEW PROPERTIES DISCOVERED IN THE B

THE DAILY GRID

Breakthrough in Grid Layouts

Modern developers are switching to CSS Grid for complex newspaper designs. The ability to span multiple columns and rows effortlessly has changed the landscape of web typography and design forever.



Web Standards Update

Semantic HTML continues to be the backbone of accessible web design. Using tags like article and section ensures that screen readers can navigate complex layouts like this one with ease.

WEATHER Expect clear skies and perfect rendering across all modern browsers this weekend.	FINANCE Flexbox stocks are steady while Grid futures are rising rapidly in the tech sector.	POLITICS The debate between internal and external CSS continues in the developer community.
---	---	---

SUBSCRIBE TO OUR NEWSLETTER

Enter your email

JOIN NOW

Experiment 3.5

Code:

```
<html><head><title>Product Landing Page</title>
<style>
:root{--bg:#f9f9f9;--card-bg:#fff;--text:#333;--header-bg:#fff;--border:#eee;}
@media(prefers-color-scheme:dark){:root{--bg:#1a1a1a;--card-bg:#2d2d2d;--text:#f9f9f9;--header-bg:#121212;--border:#444;}}
html{scroll-behavior:smooth;}
body{font-family:'Helvetica Neue',Arial,sans-serif;margin:0;padding:0;background-color:var(--bg);color:var(--text);transition:background 0.3s,color 0.3s;}
#header{position:fixed;top:0;width:100%;height:70px;background-color:var(--header-bg);display:flex;justify-content:space-between;align-items:center;box-shadow:0 2px 5px rgba(0,0,0,0.1);z-index:1000;}
#header-img{height:40px;margin-left:20px;filter:invert(var(--invert,0));}
@media(prefers-color-scheme:dark){:root{--invert:1;}}
#nav-bar{margin-right:20px;}
.nav-link{text-decoration:none;color:var(--text);margin:0 15px;font-weight:bold;}
.nav-link:hover{color:#007bff;}
#hero{margin-top:100px;text-align:center;padding:40px 20px;}
#form{display:flex;flex-direction:column;align-items:center;gap:10px;}
#email{padding:10px;width:300px;border:1px solid var(--border);border-radius:4px;background:var(--card-bg);color:var(--text);}
#submit{padding:10px 30px;background-color:#007bff;color:#fff;border:none;border-radius:4px;cursor:pointer;font-weight:bold;}
.container{max-width:1000px;margin:0 auto;padding:40px 20px;}
#video-container{display:flex;justify-content:center;margin:40px 0;}
#video{width:100%;max-width:560px;aspect-ratio:16/9;border:none;}
.flex-grid{display:flex;flex-wrap:wrap;gap:20px;justify-content:center;}
.card{flex:1 1 280px;max-width:320px;background:var(--card-bg);padding:30px;border-radius:8px;box-shadow:0 2px 10px rgba(0,0,0,0.2);text-align:center;border:1px solid var(--border);}
.price{font-size:2.5rem;font-weight:bold;margin:20px 0;}>
.btn-select{display:inline-block;padding:10px 20px;background:#007bff;color:#fff;text-decoration:none;border-radius:4px;}>
@media (max-width:600px){
#header{flex-direction:column;height:auto;padding:10px 0;}>
#hero{margin-top:140px;}}>
</style></head><body><header id="header"><nav id="nav-bar"><a class="nav-link" href="#features">Features</a><a class="nav-link" href="#how-it-works">How It Works</a><a class="nav-link" href="#pricing">Pricing</a></nav></header><main><section id="hero"><h1>Master Your Workflow</h1><form id="form" action="https://www.freecodecamp.org/email-submit"><input name="email" id="email" type="email" placeholder="Enter your email address" required><input id="submit" type="submit" value="Get Started"/></form></section><div class="container"><section id="features"><div class="flex-grid"><div class="card"><h3>Fast Performance</h3><p>Industry-leading speeds that keep you ahead of the curve.</p></div><div class="card"><h3>Secure Cloud</h3><p>Protected with enterprise-grade security.</p></div></div></section><section id="how-it-works"><div id="video-container"><iframe id="video" src="https://www.youtube-nocookie.com/embed/y8Yv4pn07qc"></iframe></div></section><section id="pricing"><div class="flex-grid"><div class="card"><h3>Personal</h3><div class="price">$10</div><a href="#" class="btn-select">Select</a></div><div class="card" style="border-color:#007bff;"><h3>Professional</h3><div class="price">$25</div><a href="#" class="btn-select">Select</a></div></div></section></main></body>
</html>
```

Output:

