

Advanced Challenges - Responsive Journal Lab

For Students Who Finish Early

These challenges will push your CSS and responsive design skills to the next level. Each challenge builds on the core lab and introduces modern web development techniques.

Challenge 1: Dark Mode Implementation ★ ★

Objective

Add a dark mode that respects user's system preferences and includes a manual toggle.

Part A: CSS-Only Dark Mode

Add this to your CSS file:

```
CSS
```

```
/* Dark Mode Styles */
@media (prefers-color-scheme: dark) {
  :root {
    --color-primary: #1a5a5a;
    --color-secondary: #0f2f2f;
    --color-accent: #2a7a7a;
    --color-background: #1a1a1a;
    --color-text: #e0e0e0;
    --color-hover: #3a3a0a;
  }

  .page-header {
    border-color: #4a9a9a;
  }

  .journal-entry {
    border-color: #2a7a7a;
    color: var(--color-text);
  }

  .friends-sidebar {
    border-color: #4a9a9a;
  }
}
```

Part B: Manual Dark Mode Toggle (JavaScript Required)

Add to your HTML (before closing `</body>`):

```
html
```

```

<button class="theme-toggle" aria-label="Toggle dark mode">
  🌙 Toggle Dark Mode
</button>

<script>
  // Dark mode toggle functionality

  const themeToggle = document.querySelector('.theme-toggle');
  const html = document.documentElement;

  // Check for saved theme preference or default to system preference
  const currentTheme = localStorage.getItem('theme') ||
    (window.matchMedia('(prefers-color-scheme: dark)').matches ? 'dark' : 'light');

  html.setAttribute('data-theme', currentTheme);

  themeToggle.addEventListener('click', () => {
    const theme = html.getAttribute('data-theme');
    const newTheme = theme === 'dark' ? 'light' : 'dark';

    html.setAttribute('data-theme', newTheme);
    localStorage.setItem('theme', newTheme);

    // Update button text
    themeToggle.textContent = newTheme === 'dark' ? '☀ Toggle Light Mode' : '🌙 Toggle Dark Mode';
  });
</script>

```

Add to your CSS:

CSS

```
/* Theme toggle button */
.theme-toggle {
  position: fixed;
  bottom: 20px;
  right: 20px;
  padding: 12px 20px;
  background: var(--color-primary);
  border: 2px solid var(--color-accent);
  border-radius: 25px;
  cursor: pointer;
  font-size: 16px;
  transition: all 0.3s ease;
  z-index: 1000;
}

.theme-toggle:hover {
  transform: scale(1.05);
  box-shadow: 0 4px 12px rgba(0,0,0,0.2);
}

/* Dark theme overrides using data attribute */
[data-theme="dark"] {
  --color-primary: #1a5a5a;
  --color-secondary: #0f2f2f;
  --color-accent: #2a7a7a;
  --color-background: #1a1a1a;
  --color-text: #e0e0e0;
  --color-hover: #3a3a0a;
}
```

Learning Goals:

- CSS custom properties (variables)
- System preference detection
- LocalStorage API
- Data attributes for state management

Challenge 2: CSS Container Queries ★ ★ ★

Objective

Use modern container queries to make components truly responsive based on their container size, not viewport size.

Implementation

CSS

```
/* Enable container queries on journal entries */
.journal-main {
  container-type: inline-size;
  container-name: journal-container;
}

/* Component responds to its container, not viewport */
@container journal-container (min-width: 500px) {
  .journal-entry__image {
    float: right;
    width: 40%;
    margin-left: var(--spacing-md);
  }

  .journal-entry__title {
    font-size: 1.8rem;
  }
}

@container journal-container (min-width: 700px) {
  .journal-entry {
    display: grid;
    grid-template-columns: 1fr 300px;
    gap: var(--spacing-lg);
  }

  .journal-entry__image {
    float: none;
    width: 100%;
    grid-column: 2;
    grid-row: 1 / 3;
  }

  .journal-entry__title {
    grid-column: 1;
  }

  .journal-entry__content {
    grid-column: 1;
  }
}
```

Learning Goals:

- Container queries vs media queries
- Component-based responsive design
- Browser feature detection

Browser Support Note: Container queries are relatively new. Check caniuse.com.

Challenge 3: Responsive Images with `srcset` ★ ★

Objective

Optimize image loading for different screen sizes and pixel densities.

Implementation

Update your HTML:

```
html


```

Explanation:

- `srcset`: Lists available image sizes
- `sizes`: Tells browser how big image will be at different breakpoints
- `loading="lazy"`: Defers loading images until they're near viewport

Learning Goals:

- Image optimization
- Performance best practices

- Lazy loading
-

Challenge 4: CSS Grid Masonry Layout ★ ★ ★

Objective

Create a Pinterest-style masonry layout for journal entries on large screens.

Implementation

```
css

@media screen and (min-width: 1024px) {
  .journal-main {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(400px, 1fr));
    gap: var(--spacing-lg);
    grid-auto-flow: dense;
  }

  /* Make some entries span 2 rows for variety */
  .journal-entry:nth-child(3n) {
    grid-row: span 2;
  }

  /* Adjust for masonry effect */
  .journal-entry {
    align-self: start;
  }
}

/* Alternative: True masonry with new CSS (experimental) */
@supports (grid-template-rows: masonry) {
  .journal-main {
    grid-template-rows: masonry;
  }
}
```

Learning Goals:

- Advanced CSS Grid techniques
- `auto-fit` vs `auto-fill`

- Feature detection with `@supports`
-

Challenge 5: Animated Page Transitions ★ ★

Objective

Add smooth animations when elements enter the viewport.

Implementation

Add to your CSS:

```
css

/* Intersection Observer Animation Setup */
.journal-entry {
  opacity: 0;
  transform: translateY(30px);
  transition: opacity 0.6s ease, transform 0.6s ease;
}

.journal-entry.is-visible {
  opacity: 1;
  transform: translateY(0);
}

/* Stagger animations for multiple entries */
.journal-entry:nth-child(1).is-visible {
  transition-delay: 0.1s;
}

.journal-entry:nth-child(2).is-visible {
  transition-delay: 0.2s;
}

.journal-entry:nth-child(3).is-visible {
  transition-delay: 0.3s;
}

.journal-entry:nth-child(4).is-visible {
  transition-delay: 0.4s;
}
```

Add JavaScript:

```
javascript

// Animate elements when they come into view
const observerOptions = {
  threshold: 0.1,
  rootMargin: '0px 0px -50px 0px'
};

const observer = new IntersectionObserver((entries) => {
  entries.forEach(entry => {
    if (entry.isIntersecting) {
      entry.target.classList.add('is-visible');
      observer.unobserve(entry.target); // Only animate once
    }
  });
}, observerOptions);

// Observe all journal entries
document.querySelectorAll('.journal-entry').forEach(entry => {
  observer.observe(entry);
});
```

Learning Goals:

- Intersection Observer API
- CSS transitions and transforms
- Performance-conscious animations

Challenge 6: Accessible Skip Navigation ★

Objective

Add keyboard navigation improvements for accessibility.

Implementation

Add to HTML (right after opening `<body>`):

```
html
```

```
<a href="#main-content" class="skip-link">Skip to main content</a>
```

Add to CSS:

```
css

/* Skip link for keyboard users */
.skip-link {
  position: absolute;
  top: -40px;
  left: 0;
  background: var(--color-primary);
  color: var(--color-text);
  padding: 8px;
  text-decoration: none;
  border-radius: 0 0 4px 0;
  z-index: 100;
}

.skip-link:focus {
  top: 0;
}
```

Update HTML main element:

```
html

<main class="journal-main" id="main-content">
```

Learning Goals:

- WCAG accessibility guidelines
- Keyboard navigation
- Focus management

Challenge 7: Performance Optimization ★★

Objective

Optimize your CSS for production with critical CSS and code splitting.

Steps

1. Identify Critical CSS (above-the-fold styles)

```
html

<style>
  /* Inline critical CSS here - mobile header and first entry */
  .page-container { width: 100%; padding: 10px; }
  .page-header { background: #A8F0F0; padding: 15px; }
  /* ... minimal styles for initial render */
</style>
```

2. Lazy load non-critical CSS

```
html

<link rel="preload" href="journal_responsive.css" as="style"
      onload="this.onload=null;this.rel='stylesheet'">
<noscript><link rel="stylesheet" href="journal_responsive.css"></noscript>
```

3. Minify CSS (use a build tool or online minifier)

4. Reduce Unused CSS (remove styles for features you're not using)

Learning Goals:

- Critical rendering path
- CSS delivery optimization
- Build tools basics

Challenge 8: CSS Variables with JavaScript ★ ★

Objective

Create a theme customizer that lets users adjust colors in real-time.

Implementation

Add to HTML:

```
html
```

```
<div class="theme-customizer">
  <h3>Customize Theme</h3>

  <label>
    Primary Color:
    <input type="color" id="primary-color" value="#A8F0F0">
  </label>

  <label>
    Secondary Color:
    <input type="color" id="secondary-color" value="#E8FBFB">
  </label>

  <label>
    Font Size:
    <input type="range" id="font-size" min="14" max="20" value="16">
    <span id="font-size-display">16px</span>
  </label>

  <button id="reset-theme">Reset</button>
</div>
```

Add JavaScript:

```
javascript
```

// Theme customizer

```
const root = document.documentElement;
const primaryColorInput = document.getElementById('primary-color');
const secondaryColorInput = document.getElementById('secondary-color');
const fontSizeInput = document.getElementById('font-size');
const fontSizeDisplay = document.getElementById('font-size-display');
const resetButton = document.getElementById('reset-theme');
```

```
primaryColorInput.addEventListener('input', (e) => {
  root.style.setProperty('--color-primary', e.target.value);
  localStorage.setItem('primaryColor', e.target.value);
});
```

```
secondaryColorInput.addEventListener('input', (e) => {
  root.style.setProperty('--color-secondary', e.target.value);
  localStorage.setItem('secondaryColor', e.target.value);
});
```

```
fontSizeInput.addEventListener('input', (e) => {
  const size = e.target.value + 'px';
  root.style.setProperty('--font-size-base', size);
  fontSizeDisplay.textContent = size;
  localStorage.setItem('fontSize', e.target.value);
});
```

```
resetButton.addEventListener('click', () => {
  root.style.removeProperty('--color-primary');
  root.style.removeProperty('--color-secondary');
  root.style.removeProperty('--font-size-base');
  localStorage.clear();
  primaryColorInput.value = '#A8F0F0';
  secondaryColorInput.value = '#E8FBFB';
  fontSizeInput.value = '16';
  fontSizeDisplay.textContent = '16px';
});
```

// Load saved preferences

```
window.addEventListener('DOMContentLoaded', () => {
  const savedPrimaryColor = localStorage.getItem('primaryColor');
  const savedSecondaryColor = localStorage.getItem('secondaryColor');
  const savedFontSize = localStorage.getItem('fontSize');

  if (savedPrimaryColor) {
```

```
root.style.setProperty('--color-primary', savedPrimaryColor);
primaryColorInput.value = savedPrimaryColor;
}

if (savedSecondaryColor) {
  root.style.setProperty('--color-secondary', savedSecondaryColor);
  secondaryColorInput.value = savedSecondaryColor;
}

if (savedFontSize) {
  root.style.setProperty('--font-size-base', savedFontSize + 'px');
  fontSizeInput.value = savedFontSize;
  fontSizeDisplay.textContent = savedFontSize + 'px';
}
});
```

Learning Goals:

- Manipulating CSS variables with JavaScript
- User preference persistence
- Form controls and event handling

Challenge 9: Responsive Tables ★★

Objective

Add a responsive comparison table to your journal (e.g., "Dog Parks Visited").

HTML Structure

```
html
```

```
<section class="comparison-table">
  <h2>Dog Parks Comparison</h2>
  <table class="responsive-table">
    <thead>
      <tr>
        <th>Park Name</th>
        <th>Location</th>
        <th>Size</th>
        <th>Rating</th>
        <th>Dog-Friendly</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td data-label="Park Name">Sunset Meadow</td>
        <td data-label="Location">Downtown</td>
        <td data-label="Size">Large</td>
        <td data-label="Rating">★ ★ ★ ★ ★</td>
        <td data-label="Dog-Friendly">Yes</td>
      </tr>
      <!-- More rows... -->
    </tbody>
  </table>
</section>
```

CSS Implementation

CSS


```
/* Default table styles */
.responsive-table {
  width: 100%;
  border-collapse: collapse;
  margin: var(--spacing-lg) 0;
}

.responsive-table th,
.responsive-table td {
  padding: 12px;
  text-align: left;
  border-bottom: 1px solid var(--color-accent);
}

.responsive-table th {
  background: var(--color-primary);
  font-weight: bold;
}

/* Mobile: Stack table into cards */
@media screen and (max-width: 768px) {
  .responsive-table thead {
    display: none; /* Hide header on mobile */
  }

  .responsive-table tr {
    display: block;
    margin-bottom: var(--spacing-md);
    border: 2px solid var(--color-accent);
    border-radius: 8px;
    padding: var(--spacing-sm);
    background: var(--color-secondary);
  }

  .responsive-table td {
    display: block;
    text-align: right;
    padding: 8px;
    border: none;
  }

  .responsive-table td:before {
    content: attr(data-label);
  }
}
```

```
float: left;
font-weight: bold;
color: var(--color-text);
}
}
```

Learning Goals:

- Responsive table patterns
- Data attributes in CSS
- Mobile-first table design

Challenge 10: Progressive Web App (PWA) ★ ★ ★ ★

Objective

Make your journal work offline as a Progressive Web App.

Steps

1. Create a manifest.json file:

```
json
```

```
{
  "name": "Victoria's Journal",
  "short_name": "Journal",
  "description": "My personal responsive journal",
  "start_url": "/",
  "display": "standalone",
  "background_color": "#A8F0F0",
  "theme_color": "#A8F0F0",
  "icons": [
    {
      "src": "icon-192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "icon-512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]
}
```

2. Link manifest in HTML:

html

```
<link rel="manifest" href="manifest.json">
<meta name="theme-color" content="#A8F0F0">
```

3. Create service worker (sw.js):

javascript

```

const CACHE_NAME = 'journal-v1';
const urlsToCache = [
  '/',
  '/index.html',
  '/basic.css',
  '/journal_responsive.css',
  '/images/victoria_with_patrick.jpg',
  '/images/sunny_beach.jpg',
  // ... other assets
];

// Install event - cache assets
self.addEventListener('install', (event) => {
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then((cache) => cache.addAll(urlsToCache))
  );
});

// Fetch event - serve from cache, fallback to network
self.addEventListener('fetch', (event) => {
  event.respondWith(
    caches.match(event.request)
      .then((response) => response || fetch(event.request))
  );
});

// Activate event - clean up old caches
self.addEventListener('activate', (event) => {
  event.waitUntil(
    caches.keys().then((cacheNames) => {
      return Promise.all(
        cacheNames.map((cacheName) => {
          if (cacheName !== CACHE_NAME) {
            return caches.delete(cacheName);
          }
        })
      );
    })
  );
});

```

4. Register service worker in your HTML:

```
javascript
```

```
if ('serviceWorker' in navigator) {  
  navigator.serviceWorker.register('/sw.js')  
    .then((registration) => {  
      console.log('Service Worker registered:', registration);  
    })  
    .catch((error) => {  
      console.log('Service Worker registration failed:', error);  
    });  
}
```

Learning Goals:

- Progressive Web Apps
- Service Workers
- Offline functionality
- App-like experience

Bonus: Testing & Validation Checklist

Automated Testing Tools

- **HTML Validation:** <https://validator.w3.org/>
- **CSS Validation:** <https://jigsaw.w3.org/css-validator/>
- **Accessibility:** <https://wave.webaim.org/>
- **Performance:** Google Lighthouse (in Chrome DevTools)
- **Responsive:** <http://responsivetesttool.com/>

Manual Testing

```
markdown
```

- [] Test on actual devices (phone, tablet, laptop)
 - [] Test with keyboard navigation only
 - [] Test with screen reader (NVDA, JAWS, VoiceOver)
 - [] Test zoomed to 200%
 - [] Test in different browsers
 - [] Test slow 3G connection
 - [] Test with images disabled
 - [] Test with CSS disabled
 - [] Print the page
 - [] Check color contrast ratios
-

Final Project Showcase Ideas

1. Add More Journal Types:

- Travel journal with maps
- Food journal with recipes
- Fitness journal with charts

2. Integrate APIs:

- Weather API for daily weather in entries
- Unsplash API for random images
- Quote API for inspirational quotes

3. Advanced Features:

- Search functionality
- Filter by date/category
- Tag system
- Comments section

Good luck with these challenges! Remember: the best way to learn is by doing. Don't be afraid to break things and experiment! 🚀