

# Quick Reference Guide: Responsive Journal Lab

## Breakpoint Strategy

### Mobile First Philosophy

Start with the smallest screen and enhance for larger displays.

Mobile (320px+) → Tablet (768px+) → Desktop (1024px+) → Large (1440px+)

Single Column      Floating Images      Two-Column Grid      Optimized Spacing

## Exercise Quick Guide

### Exercise 1: HTML Structure (20 min)

**Goal:** Add semantic HTML5 elements and BEM classes

**Checklist:**

- ☐ Replace `<div>` with `<header>`, `<main>`, `<article>`, `<aside>`
- ☐ Add viewport meta tag
- ☐ Use BEM naming: `.block__element--modifier`
- ☐ Add proper alt text to images
- ☐ Add student name and date as HTML comment

**Expected Borders (Exercise 1):**

- Header: 5px solid red
- Journal entries: 5px solid blue
- Friends sidebar: 5px solid green
- Subtitle: 5px solid purple

### Exercise 2: Spacing & Backgrounds (15 min)

**Goal:** Add padding, margins, and backgrounds

**Mobile (Default):**

css

```
.page-container {  
  width: 100%;  
  padding: 10px;  
}  
  
.journal-entry {  
  background: #E8FBFB;  
  padding: 5px;  
  margin-top: 10px;  
}  
  
.friends-sidebar {  
  background: white;  
}
```

### Tablet (768px+):

```
css  
  
@media screen and (min-width: 768px) {  
  .page-container {  
    margin: 0 10%;  
    width: 80%;  
  }  
}
```

### Desktop (1024px+):

```
css  
  
@media screen and (min-width: 1024px) {  
  .page-container {  
    background-image: url('background.gif');  
  }  
}
```

---

## Exercise 3: Float & Clear (15 min)

**Goal:** Float images and align header

**Key Concepts:**

CSS

```
/* Tablet and up */
@media screen and (min-width: 768px) {
  .journal-entry__image {
    float: right;
    width: 45%;
    margin-left: 15px;
    margin-bottom: 15px;
  }

  /* Clearfix to contain floats */
  .journal-entry::after {
    content: "";
    display: table;
    clear: both;
  }

  .page-header__title {
    text-align: right;
  }
}
```

**Why Clearfix?** Floated elements escape their containers. Clearfix forces the container to wrap around them.

---

## Exercise 4: Finishing Touches (10 min)

**Goal:** Update borders and styling

**Changes:**

CSS

```
/* Friends sidebar */
.friends-sidebar {
  border: 10px solid white;
}

/* Journal entries */
.journal-entry {
  border: 4px solid #C2E9E9;
}

/* Page header subtitle */
.page-header__subtitle {
  border-bottom: 2px dashed blue;
  font-size: 14pt;
}

/* Remove other borders */
}

/* Page header */
.page-header {
  background: #A8F0F0;
  border: none; /* Remove border */
}
```

## Exercise 5: Two-Column Layout (Advanced)

**Goal:** Add friends sidebar as left column

**Desktop Grid Layout:**

CSS

```
@media screen and (min-width: 1024px) {  
  .page-container {  
    display: grid;  
    grid-template-columns: 250px 1fr;  
    grid-template-areas:  
      "header header"  
      "sidebar main";  
    gap: 20px;  
    max-width: 1200px;  
    margin: 0 auto;  
  }  
  
  .page-header { grid-area: header; }  
  .friends-sidebar { grid-area: sidebar; }  
  .journal-main { grid-area: main; }  
}
```

## Exercise 6: Hover Effects (Advanced)

**Goal:** One selector for all hovers

**The Challenge:**

```
CSS  
  
/* Method 1: Multiple selectors */  
.journal-entry:hover,  
.friends-sidebar:hover,  
.page-header:hover {  
  background-color: lightyellow;  
}  
  
/* Method 2: ONE selector (the challenge!) */  
* {  
  transition: background-color 0.3s ease;  
}  
  
*:hover {  
  background-color: lightyellow;  
}
```

**Why it works:** The universal selector `*` targets all elements. The `:hover` pseudo-class applies when any element is hovered.

---

## 🔧 Common Issues & Solutions

### Issue 1: Images Overflowing Containers

**Problem:** Floated images extend beyond their containers

**Solution:**

```
css

journal-entry::after {
  content: "";
  display: table;
  clear: both;
}
```

### Issue 2: Layout Breaking on Mobile

**Problem:** Desktop styles affecting mobile

**Solution:** Always write mobile-first. Desktop styles should be INSIDE media queries.

```
css

/* ✅ CORRECT */
.container { width: 100%; } /* Mobile */
@media (min-width: 768px) {
  .container { width: 80%; } /* Tablet+ */
}

/* ❌ WRONG */
.container { width: 80%; } /* Applies to mobile too! */
```

### Issue 3: Content Too Wide on Large Screens

**Problem:** Text stretches across entire 4K monitor

**Solution:**

```
css
```

```
.page-container {  
  max-width: 1200px;  
  margin: 0 auto; /* Centers the container */  
}
```

## Issue 4: Viewport Not Working

**Problem:** Page not responsive on mobile

**Solution:** Add to `<head>`:

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

## Issue 5: Borders Not Showing

**Problem:** Can't see borders after Exercise 4

**Solution:** Check that you're using the correct selectors and haven't accidentally removed borders from wrong elements.



## Responsive Testing Checklist

### Screen Sizes to Test

- ✓ 320px - iPhone SE (smallest)
- ✓ 375px - iPhone 12/13
- ✓ 768px - iPad portrait
- ✓ 1024px - iPad landscape / small laptop
- ✓ 1440px - Desktop
- ✓ 1920px - Large desktop

## Testing Tools

### Browser DevTools:

1. Open Chrome DevTools (F12)
2. Click device toolbar icon (Ctrl+Shift+M)
3. Select device or enter custom dimensions

4. Toggle between portrait/landscape

Responsive Design Mode (Firefox):

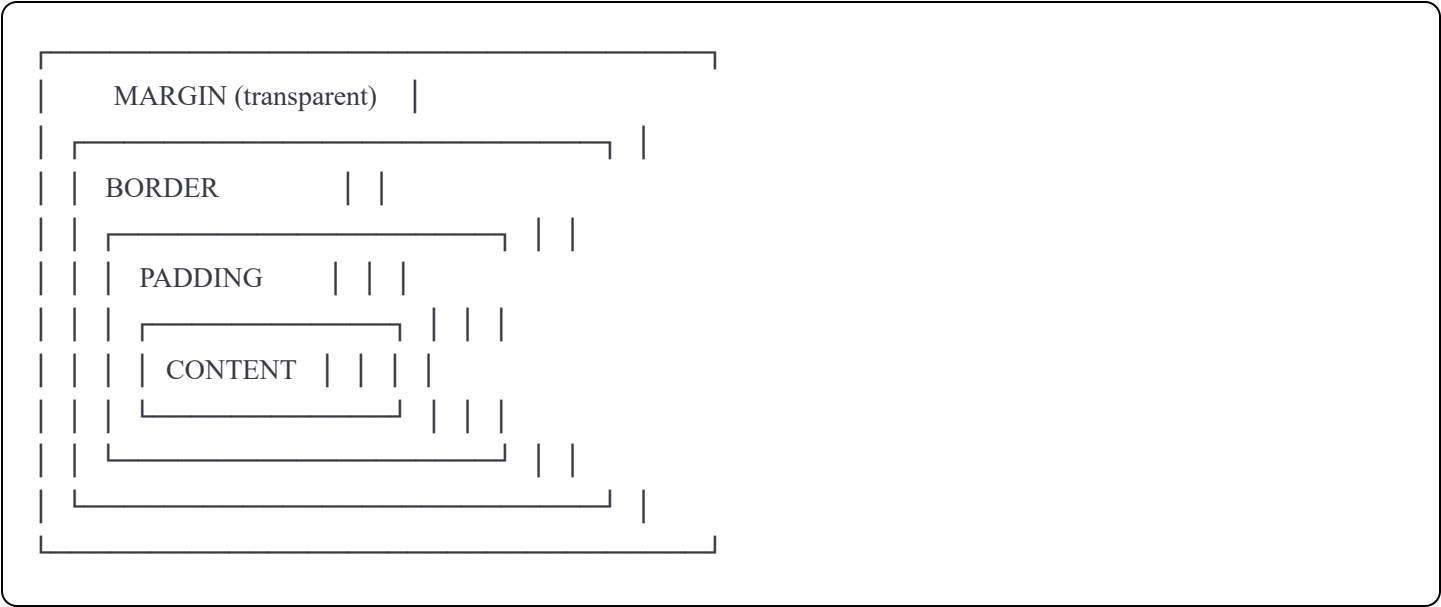
- 1. Press Ctrl+Shift+M
- 2. Choose device or custom size
- 3. Take screenshots at each breakpoint

🎨 CSS Properties Reference

Layout Properties

Property	Purpose	Example
display: grid	Create grid layout	Two-column layout
display: flex	Create flexbox	Center content
float: right	Float element	Image beside text
clear: both	Clear floats	Prevent overlap
position: relative/absolute	Positioning	Overlays

Box Model



Spacing Properties

CSS



```
/* All sides */
```

```
padding: 10px;
```

```
margin: 10px;
```

```
/* Vertical | Horizontal */
```

```
padding: 10px 20px;
```

```
/* Top | Right | Bottom | Left (clockwise) */
```

```
padding: 10px 20px 15px 5px;
```

```
/* Individual sides */
```

```
padding-top: 10px;
```

```
margin-left: 20px;
```

---

## Debugging Tips

### 1. Border Boxing Method

Add temporary borders to see layouts:

```
CSS
```

```
* {  
  border: 1px solid red !important;  
}
```

### 2. Background Color Method

Use different colors to see sections:

```
CSS
```

```
.page-header { background: pink; }  
.journal-main { background: lightblue; }  
.friends-sidebar { background: lightgreen; }
```

### 3. Chrome DevTools Grid Inspector

1. Select grid container in Elements panel
2. Click "grid" badge
3. See grid lines overlay

## 4. Check Computed Styles

1. Select element in DevTools
  2. Go to "Computed" tab
  3. See final applied styles
- 

### Pro Tips

#### Tip 1: Use CSS Variables

```
css

:root {
  --spacing: 10px;
  --primary-color: #A8F0F0;
}

.container {
  padding: var(--spacing);
  background: var(--primary-color);
}
```

#### Tip 2: Mobile-First Media Queries

```
css



/* Default: Mobile */
.container { font-size: 16px; }

/* Tablet and up */
@media (min-width: 768px) {
  .container { font-size: 18px; }
}

/* Desktop and up */
@media (min-width: 1024px) {
  .container { font-size: 20px; }
}
```

### Tip 3: Keep Selectors Simple

CSS

```
/*  Too specific */  
.page-container .journal-main .journal-entry .journal-entry__title { }  
  
/*  Better */  
.journal-entry__title { }
```

### Tip 4: Group Related Styles

CSS

```
/* Colors together */  
.journal-entry {  
  background: #E8FBFB;  
  border-color: #C2E9E9;  
  color: #333;  
}  
  
/* Spacing together */  
.journal-entry {  
  padding: 10px;  
  margin: 15px 0;  
}
```

### Tip 5: Comment Your Code

CSS

```
/* =====  
Journal Entries  
===== */  
  
/* Individual entry container */  
.journal-entry {  
  /* Layout */  
  display: block;  
  
  /* Spacing */  
  padding: 10px;  
  margin: 15px 0;  
  
  /* Visual */  
  background: #E8FBFB;  
  border: 4px solid #C2E9E9;  
}
```

---

## Submission Checklist

Before submitting your lab:

- ☐ Name and date in HTML comment
- ☐ All 6 exercises completed
- ☐ Code validated (HTML & CSS)
- ☐ Tested at all breakpoints
- ☐ Comments added for clarity
- ☐ No console errors
- ☐ Images have alt text
- ☐ Code is properly indented
- ☐ Files named correctly
- ☐ Works in Chrome, Firefox, Safari

---

## Next Steps After Lab

1. **Portfolio Project:** Convert your own journal into a responsive site
2. **Blog Template:** Create a responsive blog layout
3. **Photo Gallery:** Build a responsive image gallery

4. **Navigation Menu:** Create a responsive hamburger menu

5. **Form Design:** Make a responsive contact form

---

## Additional Resources

### Documentation

- [MDN Web Docs - Responsive Design](#)
- [CSS Tricks - Complete Guide to Grid](#)
- [W3C - Mobile Web Best Practices](#)

### Tools

- [Can I Use](#) - Browser compatibility
- [Chrome DevTools Guide](#)
- [Responsively App](#) - Multi-device testing

### Practice

- [CSS Diner](#) - CSS selector practice
  - [Flexbox Froggy](#) - Flexbox game
  - [Grid Garden](#) - CSS Grid game
- 

## Time Management

Exercise	Estimated Time	Focus
Exercise 1	20 min	HTML structure
Exercise 2	15 min	Spacing & colors
Exercise 3	15 min	Float & alignment
Exercise 4	10 min	Polish
Exercise 5	Advanced	Grid layout
Exercise 6	Advanced	Hover effects
<b>Total</b>	<b>60+ min</b>	Complete lab

**Pro Tip:** If you finish early, work on the advanced challenges to deepen your understanding!

---

**Good luck! Remember: Making mistakes is part of learning. Don't be afraid to experiment! 🧠✨**