

Mega Reactive Web Dev Cheat Sheet — Mobile-First

Quick reference of responsive & reactive hacks, patterns, and snippets for HTML/CSS/JS developers.

Core Philosophy

Start mobile-first, favor simple, accessible patterns. Build for smallest screens, then scale up with media queries and progressive enhancement.

Viewport Meta

`<meta name="viewport" content="width=device-width, initial-scale=1.0">` — required for correct mobile scaling.

Breakpoints (suggested)

Common breakpoint examples:

- Mobile small: 320px
- Mobile large: 480px
- Tablet: 768px
- Small laptop: 1024px
- Desktop: 1280px+

Use mobile-first min-width media queries.

Units & Sizing

Preferred units:

- rem for type & consistent spacing
- % for flexible containers
- vw / vh for full-bleed sections
- clamp() for fluid type: `font-size: clamp(14px, 2.5vw, 18px);`

Layout Patterns

Flexbox and Grid basics, sticky footers, centered containers, and aspect-ratio hacks. Examples included below.

Flexbox essentials

Center anything:

`display:flex; justify-content:center; align-items:center;`

Column on mobile, row on desktop using media queries.

Grid essentials

Mobile single column to multi-column on wider screens:

`display:grid; grid-template-columns: 1fr; gap:1rem;`

`@media (min-width:768px) grid-template-columns: repeat(3,1fr);`

Responsive Nav (hamburger)

Pattern: show a toggle button on small screens, full nav on larger screens. Use `display:none` or class toggles.

Responsive Images

Use srcset & sizes and `img{width:100%;height:auto;object-fit:cover;}`. Prefer WebP/AVIF for better compression.

Common Hacks — Quick list

1. Max-width + auto margins (center wrappers)
2. Aspect-ratio via padding-top (16:9 → 56.25%)
3. Prevent iOS zoom by ensuring inputs have 16px font-size
4. 100svh / dvh fallbacks for full-height sections
5. overflow-x:hidden to stop horizontal scroll
6. -webkit-tap-highlight-color: transparent
7. touch-action: manipulation to reduce touch delay
8. env(safe-area-inset-*) for notches
9. line-clamp with -webkit-line-clamp for truncation
10. intersection observer for on-scroll animations

Detailed Hacks — with snippets

1) Max Width + Auto Margin

```
.wrapper { width: min(100%, 1200px); margin-inline: auto; padding: 1rem; }
```

2) Aspect Ratio (pure CSS)

```
.aspect-169 { position: relative; padding-top: 56.25%; } .aspect-169 > * { position: absolute; inset:0; }
```

3) 100vh mobile fix

```
height: 100svh; /* fallback: 100dvh; 100vh */
```

4) Prevent iOS input zoom

```
input, select, textarea { font-size: 16px; }
```

5) Hide FOUC

```
html { visibility: hidden; } html.loaded { visibility: visible; }
```

```
window.addEventListener('DOMContentLoaded',()=>document.documentElement.classList.add('loaded'));
```

6) Sticky footer (flexbox)

```
body{min-height:100vh;display:flex;flex-direction:column;} footer{margin-top:auto;}
```

7) Clickable card (no JS needed)

```
.card{position:relative;} .card a{position:absolute; inset:0; z-index:1;}
```

8) Equal-width with flex

```
.item{flex:1 1 0}
```

9) Safe-area insets

```
padding-bottom: env(safe-area-inset-bottom);
```

10) Line clamp

```
.line-2{display:-webkit-box;-webkit-line-clamp:2;-webkit-box-orient:vertical;overflow:hidden;}
```

Reactive JavaScript Patterns

Resize listener (use sparingly):

```
window.addEventListener('resize', ()=>{ if(window.innerWidth < 768){ /* mobile code */ } });
```

Prefer matchMedia and passive listeners, and use IntersectionObserver instead of scroll-heavy code.

Performance & Accessibility hacks

- Prefer transforms over position changes for animation (transform: translateX())
- Use prefers-reduced-motion media query
- Ensure 44px+ tap targets
- Use semantic HTML (buttons, labels) for accessibility
- Use rel=preload for critical fonts

Animations & GPU

Force composition with transform: translateZ(0) for smoother animations, but avoid overusing it. Use will-change sparingly.

CSS Utilities & Variables

Use CSS variables for theming and responsive spacing:

```
:root{--space:clamp(.5rem,2vw,2rem);} section{padding:var(--space);}
```

Dev Tools & Testing

Chrome DevTools mobile emulator, Lighthouse audits, WebPageTest, cover various real devices if possible. Test on Safari iOS and Chrome Android.

Printable checklist

- Viewport meta included
- Mobile-first base styles
- Typography uses rem/clamp
- Images use srcset
- No horizontal overflow
- Tap targets >=44px
- Accessible forms
- Keyboard focus states
- Performance: compressed images, critical CSS
- Lighthouse score goals: P 90+, A11y 90+

Footer / Next Steps

This PDF and HTML contain the full mega cheat sheet. Use the HTML file to view interactively; use the PDF for printing or offline reference. See the HTML's top nav to jump to sections.

Short Table of Contents

1	Title
2	Core Philosophy
3	Viewport Meta
4	Breakpoints (suggested)
5	Units & Sizing
6	Layout Patterns
7	Flexbox essentials
8	Grid essentials
9	Responsive Nav (hamburger)
10	Responsive Images
11	Common Hacks — Quick list
12	Detailed Hacks — with snippets
13	Reactive JavaScript Patterns
14	Performance & Accessibility hacks
15	Animations & GPU
16	CSS Utilities & Variables
17	Dev Tools & Testing
18	Printable checklist
19	Footer / Next Steps