STRETCHLEARN · WORKBOOK

Responsive Design Principles - Workbook

This workbook turns the Responsive Design Principles course into one real, finished multi-breakpoint layout - a landing or dashboard screen specified at mobile, tablet, and desktop. Each section pairs with a course module and mixes guided exercises, fill-in worksheets, and checklists so you end up with an actual breakpoint map, a fluid type and spacing scale, and per-component reflow rules a developer can build - not just notes. Work through it with a design tool open (Figma, Penpot, or Affinity Designer), a browser with dev-tools device mode, and a real piece of content to lay out. The value is in the specification you produce: the widths, units, and reflow decisions written down so one layout works from a 320px phone to a 1440px desktop.

The Responsive Mindset and Mobile-First

Set up the responsive canvas and lock in a mobile-first content order before any layout is drawn.

Worksheet: Project and Canvas Setup

Define what you are designing and the artboards it will live on before placing anything. Pick one real screen (a marketing landing page or a simple dashboard) and let it drive these choices. Use CSS-pixel widths, not hardware resolutions.

Screen being designed (e.g. SaaS landing page / analytics dashboard)

Primary user goal on this screen (the one action that matters most)

Mobile artboard width (e.g. 375px; note if also checking 360 / 320)

Tablet artboard width (e.g. 768px portrait)

Desktop artboard width (e.g. 1440px) and content max-width (e.g. 1140-1280px)

Column grid per breakpoint (e.g. 4 / 8 / 12) and gutter (e.g. 16-24px)

Exercise: Decide the Mobile Content Priority

On the 375px artboard, list every block of content this screen contains, then force it into a single vertical column in priority order. This stacking order is the most important decision in the whole layout - get it right here and every wider screen becomes an arrangement of an order you trust.

• What are all the content blocks on this screen, and in what single-column top-to-bottom order do they stack on mobile?

• Which one element is the primary action, and is it within easy thumb reach (lower/centre of the screen)?

• What, if anything, is genuinely not needed on mobile - and are you sure a user does not need it?

Checklist: Mobile-First Setup Check

[ ] Meta viewport mindset confirmed: designing in CSS pixels, not hardware pixels
[ ] Mobile (375px) artboard created and designed first, before tablet or desktop
[ ] Artboards arranged narrow-to-wide, left-to-right, matching the min-width cascade
[ ] Auto Layout / constraints applied so resizing a frame tests the design
[ ] A column grid (4 / 8 / 12) is switched on over every artboard
[ ] Single-column content priority order written down for mobile

Breakpoints and the Fluid Grid

Find breakpoints from where your content breaks, make the grid fluid between them, and document the system in a breakpoint map.

Exercise: Find Your Content Breakpoints

Start at 360-375px with the mobile layout finished and slowly drag the frame wider. Each time something looks wrong - the text measure passes ~75 characters, a button stretches absurdly wide, or whitespace yawns - stop and record that width. Keep going to find the next break.

• At what pixel width did the first content break appear, and what specifically looked wrong?

• What rule fixes it (new column, larger type, repositioned element), and where is the next break after that?

• Do your content breakpoints land near common ranges (~600 / ~768 / ~1024 / ~1280px) as a sanity check?

Worksheet: Fluid Grid and Max-Width Plan

Specify how the layout flexes between breakpoints and where it stops growing. Decide column behaviour in relative terms (percentages, fr units, or auto-fit minmax) and the ceiling that keeps text readable on wide monitors.

Grid technique (percentages / CSS Grid fr units / auto-fit minmax)

Card or content grid: minimum item width for auto-fit (e.g. 250-280px)

Main proportional split if any (e.g. content 2fr / sidebar 1fr)

Content container max-width (e.g. 1140 / 1200 / 1280px)

Behaviour above max-width (centred with balanced margins?)

Outer page margin per breakpoint (e.g. 16 / 24 / auto)

Worksheet: Breakpoint Map

Build the table that turns your artboards into instructions. Name each breakpoint, give it an exact min-width value, and state what the layout does there. Use this map as the spine of your developer handoff.

Breakpoint name + min-width (e.g. sm = 0 / md = 768px / lg = 1024px / xl = 1280px)

Column count and outer margin at each breakpoint

What changes there (nav form, sidebar appears, hero splits, etc.)

What shows or hides at each breakpoint

Behaviour at real test widths 360 / 390 / 768 / 820 / 1024 / 1280 / 1440

Any gaps or overlaps found (every width 320+ has exactly one behaviour?)

Checklist: Breakpoint and Grid Check

[ ] Breakpoints chosen from where content breaks, not from device names
[ ] Each breakpoint recorded with a name and exact min-width value
[ ] Grid is fluid between breakpoints (percentages / fr / auto-fit), not fixed pixels
[ ] A content max-width holds line length on wide screens
[ ] Layout checked at 820px and other in-between widths for dead zones
[ ] Breakpoint map covers every width from 320px up with one behaviour each

Responsive Typography, Spacing, and Images

Build relative-unit type and spacing scales and define how each key image adapts in size, resolution, and crop.

Worksheet: Fluid Type Scale

Define a modular type scale in rem and decide which sizes scale fluidly with clamp(). Pick a ratio, set a base, and compress for mobile. Record min and max for any clamped headline so type handles every width itself.
Base font size (e.g. 16px = 1rem) and scale ratio (e.g. 1.25 major third / 1.333 perfect fourth)

Resulting step sizes in rem (e.g. 1 / 1.25 / 1.563 / 1.953 / 2.441rem)

Mobile vs desktop scale differences (which headlines compress on phone?)

Clamped headline spec: clamp(min, preferred-vw, max) - e.g. clamp(1.75rem, 4vw, 3rem)

Body text line-length target in ch (e.g. 60-70ch)

Line-height / leading for body and for headings

Worksheet: Spacing Scale and Touch Targets

Build a spacing ladder on a 4px or 8px base and set how section padding scales across breakpoints. Then lock the minimum sizes for every interactive element so the mobile layout is comfortable to tap.
Base spacing unit (4px or 8px) and the ladder (e.g. 4 / 8 / 16 / 24 / 32 / 48 / 64px)

Section padding per breakpoint (e.g. 24px mobile / 48px tablet / 80px desktop)

Minimum tap-target size used (44x44 Apple HIG / 48x48 Material / 44x44 WCAG 2.5.5)

Minimum spacing between adjacent tap targets (e.g. 8px)

Input/button minimum height (>= 44-48px)

Where primary actions sit for thumb reach on mobile

Exercise: Plan Each Image's Responsive Behaviour

List the key images on the screen (hero, feature, thumbnails) and decide, for each, how it adapts. Most need only fit-and-resolution handling; a few hero or feature images need art direction with a different crop per breakpoint.

• Which images simply scale with max-width 100 percent and need srcset resolution variants (list the export widths, e.g. 480 / 960 / 1440)?

• Which images need art direction - a different crop per breakpoint - and what aspect ratio at mobile / tablet / desktop (e.g. 1:1 / 4:3 / 16:9)?

• Are you exporting modern formats (WebP / AVIF) to cut file size 25-50 percent for mobile load times?

Checklist: Content Responsiveness Check

[ ] Type and spacing set in relative units (rem), respecting user font-size settings
[ ] A modular type scale is defined and reused, with mobile compression noted
[ ] Headlines that scale use clamp() with a safe min and max recorded
[ ] Body line length held to ~60-70ch
[ ] Tap targets meet 44-48px minimum with adequate spacing
[ ] Every key image has a defined fit, resolution (srcset), and crop strategy

Component Reflow and Handoff

Specify how each real component reflows across breakpoints, test the full width range, and assemble the developer handoff.

Worksheet: Component Reflow Spec

For each major component, name the responsive pattern it uses and describe its state at mobile, tablet, and desktop. This is the part of the handoff that lives in the gaps the artboards do not show, so be explicit.

Navigation: pattern (e.g. Off Canvas hamburger below 768px / priority-plus / bottom tab bar) and links shown at each breakpoint

Card grid: columns at mobile / tablet / desktop (e.g. 1 / 2 / 3-4) and auto-fit minimum width

Hero: layout per breakpoint (stacked on mobile, split two-column on desktop?)

Overall pattern mix (Mostly Fluid / Column Drop / Layout Shifter / Tiny Tweaks / Off Canvas)

What repositions vs what only resizes across breakpoints

Any element that shows/hides and the reason it is safe to hide

Exercise: Solve the Table and the Form

Take the two hardest components and commit to a strategy for each on a narrow phone, rather than letting them overflow or split. Tables get a deliberate pattern; forms go single-column with proper input sizing.

• For any data table, which strategy fits the user's need - horizontal scroll, collapse-to-cards, prioritise/hide columns, or pivot - and why?

• Is every form single-column with labels above fields, and are any side-by-side fields genuinely related short pairs (city + postal code)?

• Do all inputs meet the 44-48px minimum height, and are correct input types set so phones show the right keyboard?

Worksheet: Testing Log Across the Range

Sweep the full width range in browser dev-tools device mode and record what you find. Test the smallest real width and the longest realistic content, and confirm touch behaviour on an actual phone where possible.

Widths swept (e.g. 320 / 360 / 390 / 768 / 820 / 1024 / 1280 / 1440) and any dead zones found

Smallest width tested (320-360px) - does the layout survive?

Longest-content test (long name, three-line card) - what broke?

Real-device check - tap targets, scrolling, on-screen keyboard behaviour

Anything important hidden on mobile that a user actually needs?

Issues found and the breakpoint/rule that fixes each

Checklist: Responsive Handoff Check

[ ] Breakpoint map (names + pixel values) included in the spec
[ ] Column grids per breakpoint documented
[ ] Type and spacing scales provided in rem, with clamp() values for fluid headlines
[ ] Per-component reflow notes written for nav, cards, table, and form
[ ] Container behaviour stated explicitly (fluid up to max-width, then centred)
[ ] Full width range tested from 320px to 1440px before handoff

Your Action Plan

1. Choose one real screen (landing page or simple dashboard) and create mobile, tablet, and desktop artboards at 375 / 768 / 1440px with 4 / 8 / 12 column grids.
2. Design the 375px mobile layout completely first, fixing the single-column content priority order and placing the primary action within thumb reach.
3. Drag the frame wider and record each content breakpoint where the design visibly breaks, adding the rule that fixes each one.
4. Make the grid fluid between breakpoints with percentages or auto-fit minmax, and set a content max-width (1140-1280px) so it centres on wide screens.
5. Write the breakpoint map: name, min-width, column count, margin, and what changes/shows/hides at each breakpoint.
6. Build a modular type scale in rem (e.g. 1.25 ratio on 16px), compress headlines for mobile, and define clamp() min/max for any fluid headline.

7. Set a 4px or 8px spacing ladder, scale section padding across breakpoints, and confirm every tap target meets 44-48px.
8. Plan each key image: max-width 100 percent fit, srcset export widths for resolution, and art-direction crops where a hero needs them.
9. Specify component reflow for navigation, card grid, hero, the data table strategy, and a single-column form with proper inputs.
10. Sweep 320-1440px in dev-tools device mode, fix dead zones and long-content breaks, then assemble the spec (breakpoint map + scales + reflow notes) for handoff.