

UI Design Skills:

Core Skills and Practice



Build job-ready UI capability across nine core skill areas



Practice deliberately with feedback: fundamentals, imitation, constraints, critique, one end-to-end project



Know the 2026 boundaries:
UI vs. UX vs. product vs. graphic vs. front-end



Skill map: visual foundations, layout, type, color, components, systems, accessibility, handoff



Assessed on before/after redesigns, component states, accessibility, handoff, and a portfolio case study



How UI Practice Has Changed by 2026

PEOPLE
PRACTICE
BETTER PRODUCTS



Candidates now show wireframes, high-fidelity screens, and a clickable prototype



Design systems are an operating layer: W3C tokens stable, 84% adoption



AI use is routine: 91% weekly, up from 54% — judgment is the differentiator

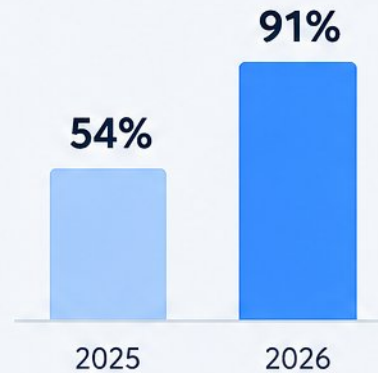


Accessibility is a legal baseline: WCAG 2.2 AA, EAA, ADA volume



Junior generalist roles shrink; specialize toward systems or research

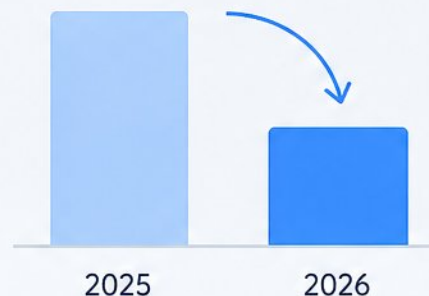
AI Tool Usage



Design Token Adoption



Junior Generalist Roles



A MORE INCLUSIVE
AND USABLE TOMORROW

Visual Foundations: Hierarchy, Layout, and Spacing



Build hierarchy with size, weight, contrast, position, and whitespace



Squint test: primary actions should still separate from secondary ones



12-column grids with 8px mini units, consistent gutters and margins



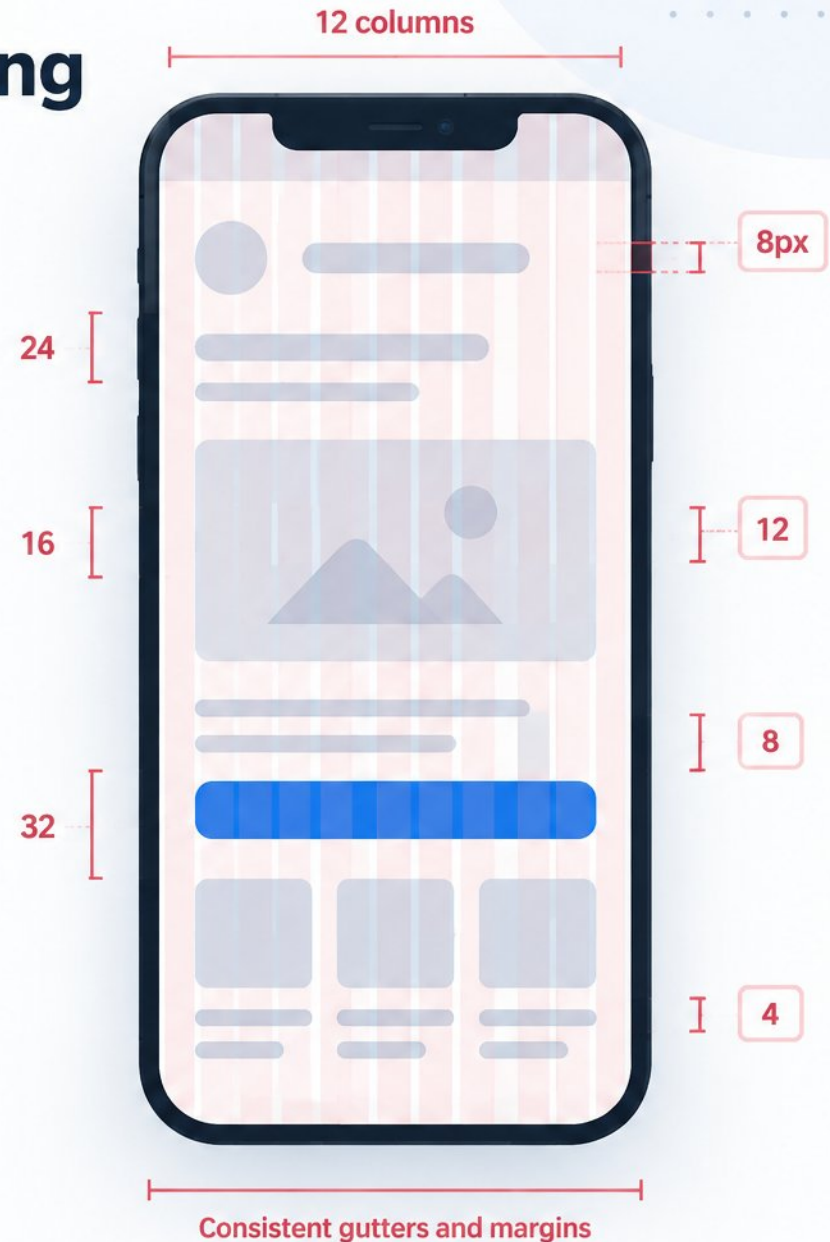
Spacing scale: 4, 8, 12, 16, 24, 32, 48, 64, 96



Responsive moves: resize, reflow, reveal, transform, divide, expand, position



Avoid crowded edges, inconsistent gutters, and shrunken desktop layouts



Typography and Color in Interface Contexts

BUILD CLEARER
INTERFACES

- 1 Type is built for UI, not posters:
base × ratio, 16px floor
- 2 Readability: line height 1.5×,
65–80 characters, paragraph
spacing 2×
- 3 WCAG 1.4.12 test: layouts must
survive text-spacing overrides
- 4 Color roles: primitives →
semantic roles → component tokens
- 5 Contrast targets: 4.5:1 body,
3:1 large and UI, APCA 75+ / 60+
- 6 OKLCH scales 50–950, contrast
grids, dark mode re-checked
not inverted
- 7 Exercise: define a type scale and
semantic palette, then grade
every text role

Type Scale

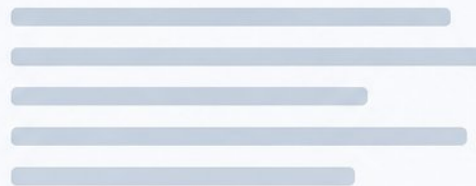
Base × ratio, 16px floor

Aa

Interface type

Built for clarity at every size.

Base size	16px floor
Scale ratios	1.125–1.618
Line height	1.5×
Measure	65–80 characters
Paragraph spacing	2×



OKLCH Color Scale

50–950 with contrast grids



Color Roles

Primitives → semantic roles → component tokens



Dark Mode Re-checked

Not inverted



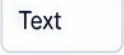
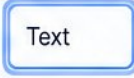
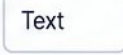
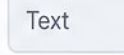
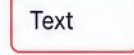
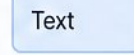
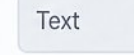
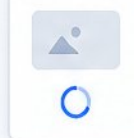
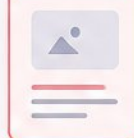
Components, States, and Interaction Design

- Build components with defined contracts: props, variants, behavior
- Map all states: default, hover, focus, active, disabled, loading, error, empty, selected
- Hover under 50ms, subtle 10% shift; use focus-visible for buttons, focus for inputs
- Keep most transitions 100–300ms; ease-out to enter, ease-in to exit
- Validate on blur, human-readable errors, preserve input, no color-only states
- Exercise: design every state of one component, document transitions and a11y notes



Transitions

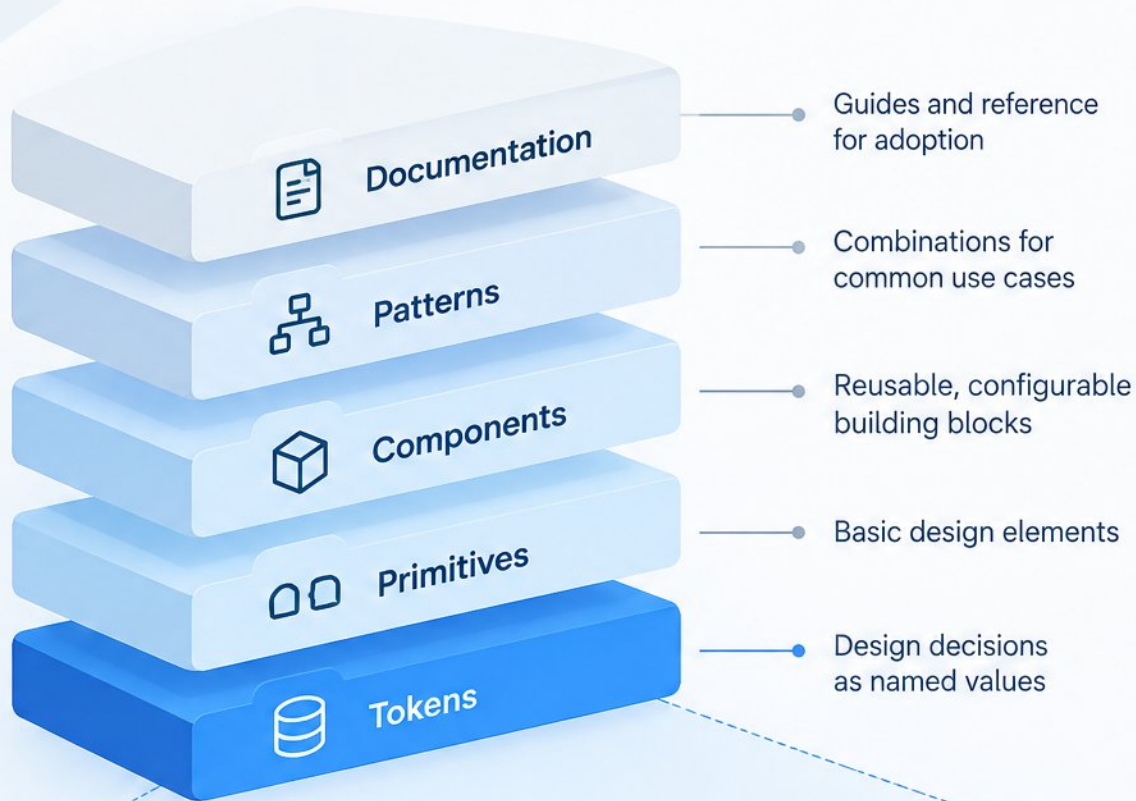
100–300ms
ease-out to enter
ease-in to exit

Component	Default	Hover	Focus	Focus-visible	Active	Disabled	Loading	Error	Empty	Selected	Read-only
Button 											
Input 											
Card 											

Focus ring contrast
3:1

Design Systems, Tokens, and Scalable Consistency

Structured pieces. Shared language. A more consistent experience.



Five layers: tokens, primitives, components, patterns, documentation



Three-layer taxonomy: primitive → semantic → component; each tier references only the one below



Name for role, not appearance: `color.action.primary`, not `button/blue`



2026 pipeline: Figma Variables → DTCG JSON → Style Dictionary → CI outputs (CSS, iOS, Android)



Component API: variant, size, and intent as the three axes



Failure modes: over-engineering, no named owner, stale docs, token drift



Exercise: audit one interface and propose token and component fixes

Three-layer taxonomy



Each tier references only the one below

2026 pipeline



Accessibility and Inclusive UI Practice as Standard Work

Design and build experiences that work for more people, by default.

- ✓ POUR frames it: perceivable, operable, understandable, robust — WCAG 2.2 AA is the 2026 baseline
- ✓ Six new 2.2 criteria: focus not obscured, dragging movements, target size, consistent help, redundant entry, accessible authentication
- ✓ Numbers to design against: 4.5:1 body text, 3:1 large text and UI, 24×24 CSS px target floor, 44×44px touch, 3:1 focus, 320px reflow
- ✓ Never signal status by color alone — pair every cue with icon, text, or pattern
- ✓ Respect prefers-reduced-motion; pause controls over 5 seconds; flashing under 3 per second
- ✓ Automated tools catch only 30–57% — tab through, zoom to 200%, spot-check a screen reader

Perceivable

People can see, hear, and detect content.

Aa

4.5:1

body text

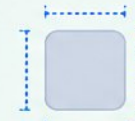
Aa

3:1

large text
and UI

Operable

People can interact with all functionality.



24×24
CSS px
target floor



44×44px
touch

Understandable

People can find, comprehend, and use information.



3:1 focus



Color alone



Color plus
icon

Robust

Content works across devices and assistive tech.



320px
reflow



200%
zoom

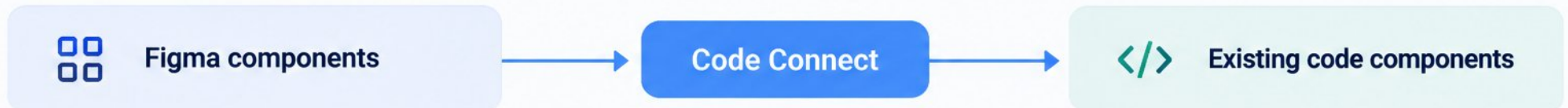
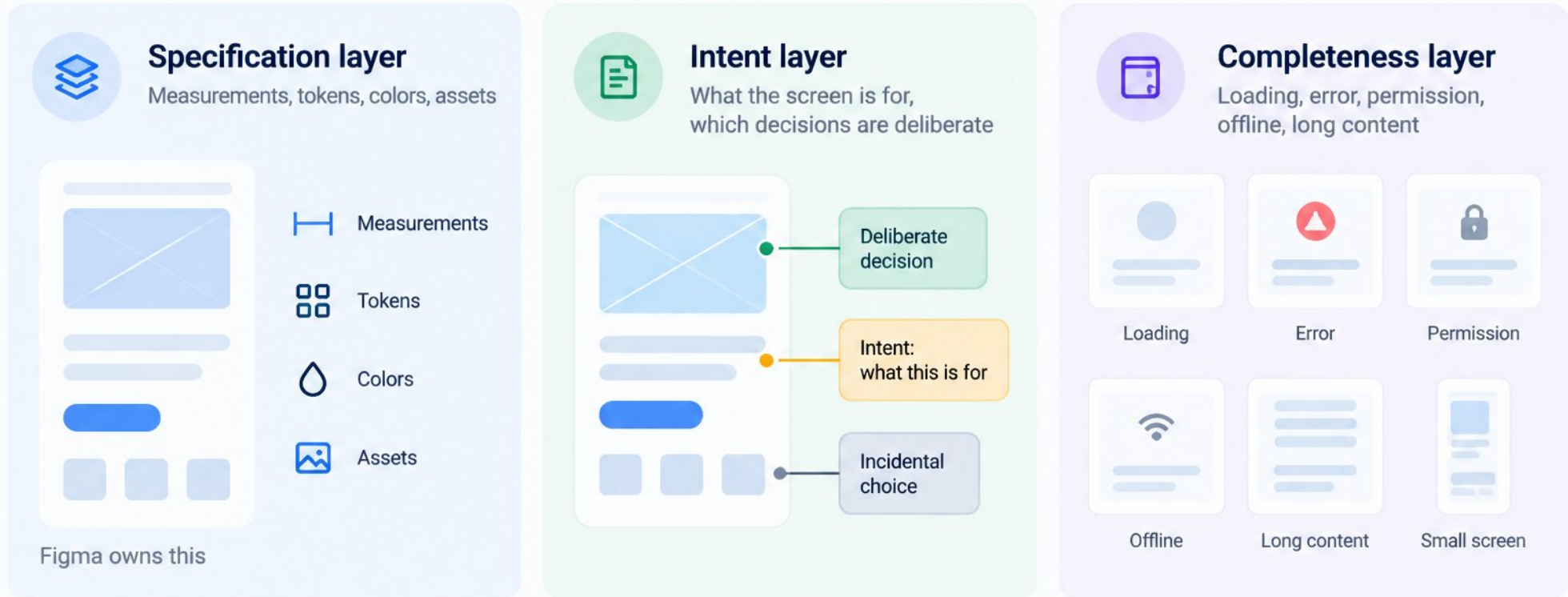


Design with people in mind.

Accessibility and inclusion aren't add-ons — they're standard work.

Tooling and Design-to-Development Handoff

From design intent to maintainable implementation

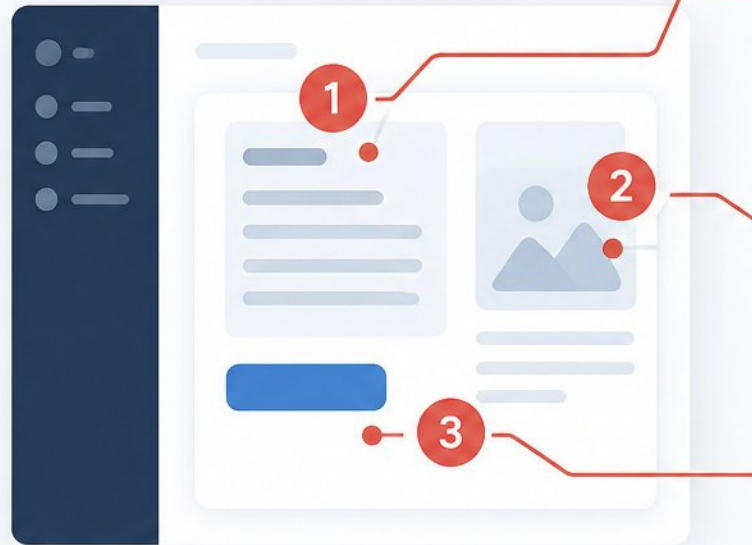


- Handoff fails on intent and missing states, not on measurements
- Specification layer: measurements, tokens, colors, assets — Figma owns this
- Intent layer: what the screen is for, which decisions are deliberate

- Completeness layer: loading, error, permission, offline, long content
- Dev Mode solved token inspection and spacing, not production code
- Code Connect maps Figma components to existing code, keeping layers honest

Critique, Testing, and Iteration

- Critique against heuristics, not taste — log issues against a named principle
- Rate severity so a mislabeled icon isn't treated like a broken payment flow
- Structure critique across roles: user experience, product goals, engineering feasibility
- Lightweight testing: testable success criterion, 3–5 participants, behavior plus reasoning
- Heuristic evaluation first, usability testing for high-risk, cognitive walkthroughs for onboarding
- Group findings by severity, frequency, and segment; fix blockers first
- Exercise: critique a peer screen, propose three evidence-based improvements



User experience and accessibility

Link each issue to a named heuristic principle.



Product and business

Consider how the issue affects product goals.



Engineering feasibility

Assess technical complexity and constraints.

Testing with real users



3–5 participants

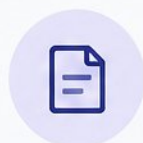
Multiple sources of evidence



Behavior



Reasoning



Support tickets

Practice Plan and Portfolio Readiness



Daily drill, weekly critique, monthly end-to-end project



Hiring managers want 3–5 deep case studies, not screen galleries



First case study screen must name the business problem and what changed



Avoid gallery portfolios, invented constraints, inflated metrics, hidden roles



Source projects from real friction, niche communities, five to eight interviews



Frame AI honestly: what you generated and corrected before shipping

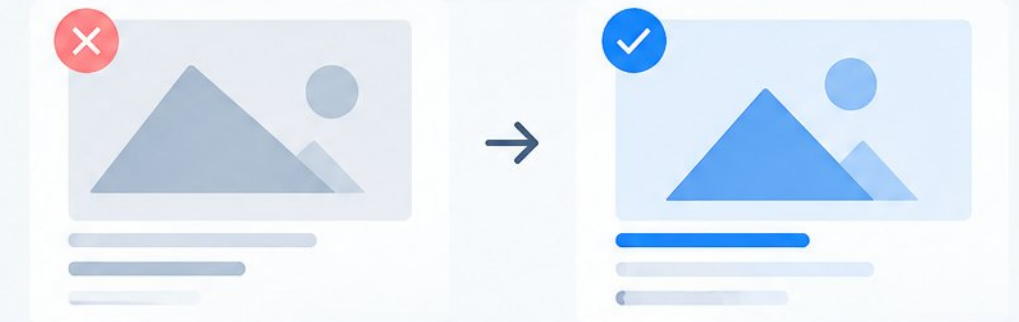


Next: 30-day plan, resource list, self-assessment checklist

Case study page anatomy



AI in the process: judgment over autopilot



30-day practice plan



PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/42597fb3/public