

# Dark Mode Design — Workbook

This workbook turns the course into a dark theme you can actually ship. Each section pairs with a course module: you choose a base surface and compute an elevation ladder from overlays, measure foreground contrast against your real surfaces and tune brand color so it stops vibrating, audit the inversion traps that flatten and glare a dark UI, then wire one semantic token set into both a light and a dark theme. Keep your filled sheets and the editable templates together, over time they become a reusable theming kit, a set of named surfaces, verified contrast pairs, and a review checklist you run on every screen rather than rediscovering each time.

## Surfaces and Elevation in the Dark

Pick a near-black base, compute an overlay-based elevation ladder, and lock the surface tones you will actually use.

### Exercise: Choose and Justify Your Base Surface

Pick a single dark base surface for your product, starting from a reference like #121212, Material's #1C1B1F, or GitHub's #0D1117. Decide whether it stays neutral gray or takes a faint brand tint at very low saturation. Paste pure white and your intended body-text gray onto it in any design tool or contrast checker and read the ratios. Confirm the base is lighter than #000000 so you keep downward headroom.

- What base hex did you choose, and is it neutral or tinted toward your brand hue (and at what rough saturation)?

---

- What contrast ratio do pure white and your body-text gray get on it, and do both clear AA (4.5:1)?

---

- Why did you reject pure black, name the specific risk (halation, no headroom, or OLED edge) that drove the decision?

---

- Does any true-black element (an AMOLED variant or media bar) still need to sit below this base?

### Worksheet: Elevation Ladder Computation

Fill one row per surface level your product needs (most need only base, card, menu, dialog). For each, record the Material overlay opacity, then composite white at that opacity over your base in a design tool and read the flattened hex. Bake each result into a named token.

Surface role (base / card / menu / sheet / dialog)

---

Elevation in dp (0 / 1 / 8 / 24, etc.)

---

White overlay opacity from the ladder (%)

---

Base hex used

---

Computed flattened surface hex

---

Token name (e.g. surface, surface-elevated-1)

---

Where it appears in the UI

---

---

### Checklist: Surface System Sanity Check

- Base surface is a near-black gray, not #000000
- Each higher surface is visibly lighter than the one behind it
- No more than four or five distinct surface tones defined (avoid indistinguishable grays)
- Every elevated surface stored as a flattened hex token, not re-stacked transparency
- Body text re-checked against the lightest elevated surface, not just the base
- A separate true-black or AMOLED variant noted as optional, not the default

## Contrast and Color on Dark Surfaces

Measure every foreground against its real surface and tune brand color until it is legible and calm.

### Exercise: Measure Text Against Every Surface

Take your high-emphasis, medium-emphasis, and disabled text colors (start from white at 87, 60, and 38 percent if you have nothing else) and flatten each to a solid hex. Using a contrast checker (Stark, WebAIM, or DevTools), measure each against your base AND each elevated surface from Section 1. Note which pairs pass AA for normal text (4.5:1), large text (3:1), and which fail.

- Which text colors passed 4.5:1 on every surface, and which dropped below it on a lighter elevated surface?

---

- Did your medium-emphasis text clear 4.5:1, or does it only pass as large text at 3:1?

---

- Your disabled text is exempt, but is it ever reused for active information that would then need to pass?

---

- What solid hex did you lock for high and medium emphasis after measuring, rather than leaving them as opacities?

### Exercise: Desaturate a Brand Color for Dark

Take one saturated brand or accent color that looks fine on white. Convert it to HSL or HSB, hold the hue, reduce saturation by 10 to 30 percent, and raise lightness until it sits calmly on your darkest surface. Test it three ways with a contrast checker: as a fill or border (needs 3:1) and as text on the surface (needs 4.5:1), plus white text on it if it is a button.

- What were the original and adapted HSL values, and how much did you drop saturation and raise lightness?

---

- Does the adapted color stop vibrating against the dark field, and does it still read as your brand?

---

- What ratio does it get as a UI border (target 3:1) and as text (target 4.5:1) on your base surface?

---

- If it is a button fill, does white label text on it pass, and did you have to lighten the fill further?

### Worksheet: Verified Color Pair Record

Record one row per foreground/background pair you certify, so engineering inherits measured values rather than guesses. Capture the exact surface, not a generic black.

Foreground role and hex (e.g. on-surface-high #E6E6E6)

---

Background surface role and hex

---

Measured contrast ratio

---

Element type (body text / large text / icon / border / button fill)

---

Required threshold (4.5:1 / 3:1)

---

Pass or fail (AA)

---

Tool used to measure

---

Notes (e.g. fails on elevated card, needs lighter value)

---

### Checklist: Color Readiness Check

- No fully saturated brand colors left on dark surfaces
- Every body-text color clears 4.5:1 on every surface it appears on
- Large text and UI boundaries clear at least 3:1
- Accent colors verified as fill, border, and text where each is used
- White-on-accent button labels measured and passing
- Disabled-only grays clearly separated from active informational grays

## Avoiding the Inversion Traps

Restore depth without light-background shadows, and handle images and assets so the theme looks deliberate.

### Exercise: Rebuild Depth Without Light-Mode Shadows

Take one card or sheet from your UI. Turn off its light-mode drop shadow and confirm it now looks flat on the dark surface. Re-create separation three ways and compare: first by relying on the lighter elevated surface tone alone, then by adding a hairline white border at about 8 to 12 percent, then by adding a darker, tighter shadow than light mode used. Decide which combination sells the elevation.

- How imperceptible was the original light-mode shadow once placed on the dark surface?
- Did the lighter surface tone alone create enough separation, or did it need help?
- What border opacity gave a crisp edge without looking like a hard outline?
- Which final combination (surface tone, border, tuned shadow) did you keep, and for which elevation level?

### Worksheet: Asset Treatment Audit

List every image, illustration, logo, and transparent PNG in the screens you are theming. For each, record how it behaves on the dark surface and the treatment it needs, scrim, recolor, dark variant, or alternate lockup.

Asset name and where it appears

---

Asset type (photo / illustration / logo / icon / transparent PNG)

---

Problem on dark (glare / hidden white edge / invisible dark strokes / none)

---

Treatment needed (dark scrim % / recolor fills / lighten strokes / alternate lockup / token-driven color)

---

Dark variant created (yes / no / n/a)

---

---

Verified on a dark screen (yes / no)

---

### Checklist: Inversion-Trap Sweep

- No reused light-mode shadows left doing nothing on dark
- Elevation reads through surface tone and borders, hierarchy is perceptible
- Large hero photos toned down with a subtle dark scrim
- Transparent PNGs checked for hidden white backgrounds and halo edges
- Illustrations have dark variants (surface-colored fills, lightened strokes)
- Logos have a light-on-dark lockup, none sit invisibly on the surface
- Icons use `currentColor` or token-driven color so they adapt per theme

## One Token Set, Two Themes

Name roles not colors, wire light and dark values into Figma modes, and ship with `prefers-color-scheme` plus a toggle.

### Exercise: Map Your Values onto Semantic Tokens

Using everything computed in Sections 1 to 3, fill out a semantic token map. For each role (surface, surface-elevated-1, on-surface, on-surface-muted, primary, on-primary, outline, error, overlay-scrim), assign the light-mode value and the dark-mode value, each pointing at a primitive from your raw palette. Keep names free of color words and component words.

- Which semantic roles does your product actually need, and did any name still sneak in a color word to rename?

---

- For surface and on-surface, what light value and dark value did each resolve to?

---

- Where did your tuned dark accent and verified text hexes land in the map?

---

- Are all your components referencing only semantic tokens, never raw hex or primitive names?

### Exercise: Build and Switch the Theme in Figma

Create a Primitives variable collection (single mode) for your raw palette, then a Semantic collection with Light and Dark modes. Add your semantic variables and alias each to the right primitive per mode. Bind one real component's fills, strokes, and text to the semantic variables, then flip the frame's mode between Light and Dark and watch it update.

- Did switching the frame mode flip the whole component correctly with no leftover hardcoded colors?

---

- Which semantic value passed in light but failed contrast in dark when you re-checked after wiring?

---

- How will you export these variables to code (Tokens Studio, the variables API), and do the names match?

---

- What broke or surprised you the first time you toggled, and how did you fix it?

### Checklist: Ship-Ready Dual-Theme Check

- Tokens organized as primitive then semantic, components bind only to semantic
- Figma Semantic collection has Light and Dark modes with aliased values
- CSS exposes semantic tokens as custom properties, dark values override under `prefers-color-scheme: dark`
- `color-scheme` set so native controls and scrollbars match the theme
- Manual toggle sets `data-theme` and persists in `localStorage`, applied before first paint

- [ ] System, Light, and Dark options offered where appropriate
- [ ] meta theme-color matches the active theme on mobile
- [ ] Module 3 review re-run in the browser, contrast verified in both modes

## Your Action Plan

1. Choose a near-black base surface and confirm body text clears AA on it, rejecting pure black for a documented reason.
2. Compute your elevation ladder by compositing white overlays at the Material opacities and bake each surface into a named token.
3. Flatten your high, medium, and disabled text colors to solid hex and measure each against every surface.
4. Desaturate and lighten every brand and accent color until it is calm and passes contrast as fill, border, and text.
5. Rebuild depth using lighter surface tone plus hairline borders and tuned shadows instead of reused light-mode shadows.
6. Audit every image, illustration, logo, and transparent PNG and create dark variants or scrimms where needed.
7. Write a semantic token map naming roles, with a light value and a dark value for each.
8. Wire the tokens into a Figma Semantic collection with Light and Dark modes and switch a real component to verify.
9. Implement the tokens as CSS custom properties with prefers-color-scheme dark overrides, color-scheme, and a persistent toggle.
10. Run the pre-ship dark-mode review on every screen and interactive state, on a real OLED screen at low brightness if possible.









