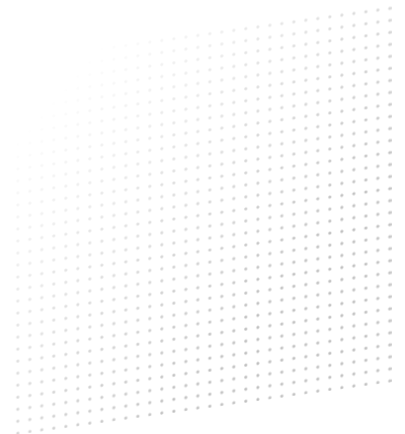

A RESOURCE FOR DESIGN SYSTEM AND ENGINEERING TEAMS

Focus ring token and usage rules.



One ring, every surface, every theme.

Tokens decide it. CSS draws it. Pixels prove it.

24

rules, each with its test

56

ring checks, light and dark, all passing

18

Tab stops measured in pixels

The ring is the keyboard user's cursor. Make it impossible to lose.

The essay this guide accompanies names focus ring visibility as the most common failure: someone writes `outline: none` because the browser ring looks wrong, and nothing replaces it. The fix is one ring, defined once as tokens, drawn by one CSS rule and checked against every surface a component can sit on.

WCAG 2.2 sets the floor: a visible indicator (SC 2.4.7), 3:1 against adjacent colours (SC 1.4.11) and never entirely hidden (SC 2.4.11), all Level AA. Focus Appearance (SC 2.4.13, AAA) asks for a 3:1 change over the area of a 2 CSS px perimeter [S01, S05]. This guide makes 2.4.13 the system target, because a design system sets it once for every product.

The kit makes each rule executable: DTCG 2025.10 focus tokens that extend Field Guide 03, a checker for seven places the ring appears on the Acme Team settings page, a CSS lint, and a Playwright script that counts the pixels that change. Field Guides 11 and 12 use the same page.

Version 1.0 / Sources checked 24 September 2026

Field Guide 10 of the Design × AI series. Verified 24 September 2026 with Node 22.22 and Playwright 1.58.2 and 1.63.0 with their bundled Chromium (identical results); token files validated against the official DTCG 2025.10 schemas with Ajv 8.

Practical guidance, not legal advice. WCAG 2.2 is a W3C Recommendation (12 December 2024); SC 2.4.13 is AAA and not required for AA conformance. Rules labelled KIT RULE go beyond WCAG. Prepared with AI assistance and edited by hand.

START HERE

Four criteria, one ring.

Map each WCAG requirement to the kit check that proves it, then pick the route that matches where your system is today.

WCAG 2.2	Level	What it asks of the ring	Kit check
2.4.7 Focus Visible	AA	Some visible indicator exists in keyboard use	<code>measure-focus.mjs</code> : pixels change
1.4.11 Non-text Contrast	AA	3:1 against the adjacent colours it is drawn over	<code>check-focus-ring.mjs</code> 1.4.11
2.4.11 Focus Not Obscured (Minimum)	AA	The focused element is not entirely hidden by author content	<code>scroll-padding</code> rule; Field Guide 12 Tab walk
2.4.13 Focus Appearance	AAA	A 3:1 change over at least a 2 CSS px perimeter's area	<code>check-focus-ring.mjs</code> 2.4.13; <code>measure-focus.mjs</code>

Minimum area for a rectangle w by h with corner radius r: $4w + 4h - (16 - 4\pi)r$. A 90 by 30 px button needs 480 px² [S05].

Starting from the browser default

Sections 01 and 03. Add the two tokens, paste `focus-ring.css`, delete every `outline: none`. Run the lint on your compiled CSS.

Auditing an existing ring

Describe it as a ring spec and run `check-focus-ring.mjs` over your surfaces. Section 02 explains each failure.

Supporting dark mode or a brand surface

R09 and R10. The ring token changes per theme; on brand surfaces the two ring colours swap.

Gating it in CI

Section 04. The Playwright spec measures the rendered ring, in normal and forced colors mode, on real routes.

Label	Meaning
TOKEN	Lives in the DTCG token files and their <code>\$description</code> .
SCRIPT	Enforced by <code>check-focus-ring.mjs</code> over the token values.
LINT	Enforced by <code>lint-focus-css.mjs</code> over compiled CSS.
BROWSER	Proven in Chromium by <code>measure-focus.mjs</code> or the Playwright spec.
WCAG AA / AAA	Traces to a WCAG 2.2 success criterion at that level.
KIT RULE	Stricter than WCAG; the reason is given.
PRACTICE	A working method with a review signal rather than a hard gate.

SECTION 01

Tokens: the ring is a decision, made once.

Four tokens describe the whole ring. Components never pick a focus colour or width; they inherit the one the system decided.

Suggested owners: Design-system lead + accessibility lead

☐ One semantic colour for the outer ring, per theme

R01 **TOKEN**

`color.border.focus` aliases `blue.600` (#125bd0) in light and `blue.300` (#80adff) in dark. Field Guide 03 already defines it: one ring in the system, not one per component.

Evidence: `build-focus-css.mjs` emits `--ds-color-border-focus` for both themes.

Field Guide 03 R05 to R08. DTCG aliases [S12].

☐ A second token for the inner band

R02 **TOKEN**

`color.border.focus-inner` (white in light, `neutral.950` in dark) sits between the component and the outer ring. Two colours that contrast with each other keep the ring visible against any fill: technique C40, and the two-tone rings Sara Soueidan recommends.

Evidence: Both focus files validate against the DTCG 2025.10 schema.

C40 two-colour indicator [S06]; Soueidan [S07].

☐ Width is a token: 2px outside, 3px when drawn inside

R03 **TOKEN** **WCAG AAA**

`stroke.focus` is 2px. `stroke.focus-inset` is 3px, because a ring drawn inside the component encloses less area: the Understanding document shows a 2px inset line failing 2.4.13 and needing 3px.

Evidence: Checker: a 2px inset ring on the 200 by 36 menu item covers 928 of the 944 px² required; 3px covers 1,380.

SC 2.4.13 Understanding, inset indicators [S05].

☐ The offset is the inner band, not a gap

R04 **TOKEN**

The outline sits `stroke.focus` away from the component and the inner band fills that space with `box-shadow`. One token governs both, and no transparent gap inherits whatever colour is behind it.

Evidence: `focus-ring.css`: `outline-offset` and the `box-shadow` spread both read `--ds-stroke-focus`.

Kit design.

☐ Each focus token says what it guarantees

R05 **TOKEN**

The `$description` names the rule the value was chosen for: what a reviewer sees when someone proposes a lighter blue, and what an agent reads before it writes a focus style.

Evidence: Field Guide 03's `check-token-rules.mjs` fails on a semantic token without `$description`.

DTCG `$description` [S12]; Field Guide 03 R10.

Nobody picks a focus colour. They inherit the one that passed.

SECTION 02

Contrast and area: the numbers the ring must reach.

A ring passes or fails where it is drawn, not in a swatch. Check it against every surface in every theme, and check it as area, not only as colour.

Suggested owners: Accessibility lead + design-system lead

☐ 3:1 against the colour the ring is drawn over

R06 **SCRIPT** **WCAG AA**

An outset ring must contrast with the surface around the component; an inset ring with the component's own fill. Ratios are not rounded: 2.99:1 fails.

Evidence: `check-focus-ring.mjs` 1.4.11 check: outer ring 5.50:1 on `bg.subtle` and 6.12:1 on `bg.surface` in light, 7.76:1 and 8.54:1 in dark.

SC 1.4.11, focus indicators [S03].

☐ The contrasting part covers a 2px perimeter's area

R07 **SCRIPT** **WCAG AAA**

Only the bands that change by 3:1 between unfocused and focused count toward the area. A white inner band on a white surface changes nothing, so the kit counts only the outer ring: 625 px² against 587 required on the 112 by 40 Send invite button.

Evidence: `check-focus-ring.mjs` 2.4.13 check, using $4w + 4h - (16 - 4\pi)r$ for the required area.

SC 2.4.13, minimum area and two-colour indicators [S05].

☐ The ring reads as a ring, not a bigger component

R08 **SCRIPT** **KIT RULE**

A 2px ring with no offset in the button's own colour passes 1.4.11 and 2.4.13 and still looks like a slightly larger button. The same happens when a blue ring touches a grey border: blue.600 against neutral.600 is 1.03:1. The kit requires the contrasting band to reach 3:1 against whatever sits between it and the component.

Evidence: Separation check: the no-offset ring fails 10 of 14 contexts, on every filled or bordered control.

Kit rule, stricter than WCAG. Soueidan on offsets [S07].

☐ Every surface in every theme is a test case

R09 **SCRIPT**

`focus-contexts.json` lists where the ring appears: page background, dialog surface, bordered field, tab list, clipped menu, brand banner. Adding a surface to the product means adding a row.

Evidence: 7 contexts by 2 themes by 4 checks: 56 of 56 pass for the kit ring.

Kit script.

☐ **On brand surfaces the two ring colours swap**R10 **TOKEN** **SCRIPT**

On the blue trial banner the blue outer ring disappears and the inner band merges with the white button.

Re-point the two tokens inside `[data-surface="brand"]`: outer ring to `color.text.on-brand`, inner band to `color.bg.brand`. No second ring palette.

Evidence: Without the swap the kit ring fails separation on the banner in both themes; with it, the outer ring is 6.12:1 in light and 8.54:1 in dark.

C40: each colour covers the backgrounds the other cannot [S06].

☐ **Where a container clips, draw inside at 3px**R11 **SCRIPT** **BROWSER** **WCAG AAA**

Menus, listboxes and grid cells often sit in `overflow: hidden` containers that cut an outset ring off completely. Use the inset variant with `stroke.focus-inset`.

Evidence: `measure-focus.mjs` on the violations demo: the clipped menu item shows 0 changed pixels; the kit demo measures 1,380 px² against 944 required.

SC 2.4.13 [S05].

☐ **Never one ring colour for both themes**R12 **SCRIPT** **WCAG AA**

A light blue chosen for dark mode reaches 2.25:1 on a white surface. The ring token is per theme like every other semantic colour.

Evidence: Checker, one-colour design (#80adff): fails 1.4.11 in all 7 light contexts and on the dark banner.

SC 1.4.11 [S03].

A ring that passes in a swatch can fail on the page. Check the page.

SECTION 03

CSS: one rule, drawn the same way everywhere.

The tokens are only as good as the CSS that draws them. These rules are what the lint reads in compiled CSS, whoever or whatever wrote it.

Suggested owners: Engineering lead + design-system lead

☐ **Style `:focus-visible`, not `:focus`**

R13 **LINT**

`:focus-visible` matches when the browser decides focus should be shown: keyboard use, and always for text inputs. A ring on `:focus` also appears on mouse click, which is how rings get deleted in design review.

Evidence: Lint FC05 warns on a ring styled on `:focus`.

MDN `:focus-visible`; Baseline since March 2022 [S09].

☐ **No outline removal without a later replacement**

R14 **LINT** **WCAG AA**

`outline: none` is allowed only when a `:focus-visible` rule for the same selector follows it. A global `:focus-visible` ring does not rescue `button:focus { outline: none }`: the button rule is more specific and wins.

Evidence: Lint FC01. The essay's own bad example, `button:focus { outline: none; }`, fails it; its mixin, which pairs the removal with a `:focus-visible` ring, passes.

SC 2.4.7 [S02]; Soueidan [S07].

☐ **The outer ring is an outline, never box-shadow alone**

R15 **LINT** **BROWSER**

Forced colors mode forces `box-shadow` to `none` and repaints outline colours with system colours. A box-shadow ring vanishes for exactly the people who chose high contrast. Keep an outline, even a transparent one.

Evidence: Lint FC02. `measure-focus.mjs --forced-colors`: kit demo 18 of 18 stops pass; violations demo 3 of 18.

MDN forced-colors [S10]; CSS Color Adjustment Level 1 [S11].

☐ **No literal colours or widths in focus rules**

R16 **LINT**

Every focus rule reads `var(--ds-color-border-focus*)` and `var(--ds-stroke-focus*)`. A literal is a second ring nobody will update.

Evidence: Lint FC03 and FC04. System colours inside `@media (forced-colors: active)` are allowed.

Field Guide 03 R15 and R16.

☐ **Give forced colors a system colour; never opt out**R17 **LINT**

Set `outline-color: Highlight` in the forced-colors media query; authored system colours are kept, other colours are replaced. `forced-color-adjust: none` on a focus rule throws away the user's palette.

Evidence: Lint FC06; `focus-ring.css` forced-colors block.

CSS Color Adjustment Level 1 [S11]; MDN [S10].

☐ **Reserve the sticky header's height**R18 **PRACTICE** **WCAG AA**

`scroll-padding-block-start` equal to the header height keeps a focused element from scrolling under it. Field Guide 12's Tab walk found three tabs entirely hidden on the broken fixture without it.

Evidence: `focus-ring.css` sets `scroll-padding-block-start`; Field Guide 12 `tab-walk.mjs` reports `obscured`.

SC 2.4.11; scroll-padding is a sufficient technique [S04].

☐ **Components cannot restyle the ring**R19 **PRACTICE**

The ring is listed as locked in each component contract, next to minimum target size. A component that needs a different ring is a token decision, reviewed once.

Evidence: `theming.locked[]` in the Field Guide 02 contract includes the focus ring; lint FC03 and FC04 catch local overrides.

Field Guide 02, item 06.02.

If the lint passes, the only ring in the CSS is the one the tokens describe.

SECTION 04

Gates: prove the ring on real pages.

Token maths catches design mistakes; pixel measurement catches the cascade, clipping and specificity. Run both.

Suggested owners: Engineering lead + accessibility lead

☐ **Run the ring checker on every token change**

R20 **SCRIPT**

Any change to a colour the ring or a surface uses re-runs the 56 checks. A new surface in the product is a new row in `focus-contexts.json` first.

Evidence: `npm run ring:check` exits 1 on any failure; the six violation designs fail 96 of 336 checks.

Kit script.

☐ **Lint compiled CSS, not source**

R21 **LINT**

Sass mixins, CSS-in-JS and utility classes all end up as CSS. Lint what ships, so a ring removed in a third-party stylesheet is caught too.

Evidence: `lint-focus-css.mjs dist/**/*.css` in CI: 0 problems on `focus-ring.css`, 9 errors and 1 warning on the violations stylesheet.

Kit lint.

☐ **Measure the rendered ring in pixels**

R22 **BROWSER** **WCAG AAA**

Tab through each key route, screenshot every stop focused and unfocused, and count the pixels that change by 3:1. This is 2.4.13 as a test, and it sees what a style audit cannot.

Evidence: `measure-focus.mjs` or `tests/focus-ring.spec.mjs`: 18 of 18 stops pass on the kit demo; 7 of 18 on the violations demo.

SC 2.4.13 [S05]; Playwright `emulateMedia` [S13].

☐ **Test forced colors, then check on Windows**

R23 **BROWSER** **PRACTICE**

Run the same measurement with `forcedColors: 'active'`. Emulation is not the same as a Windows contrast theme, so a person checks the key routes on Windows once per release.

Evidence: The spec runs both modes; the release record names the Windows build and theme used.

Playwright [S13]; CSS Color Adjustment [S11].

☐ **Record what a human checked**

R24 **PRACTICE**

The scripts cover contrast, area, visibility and forced colors. A person still judges whether the ring is easy to find at 200% zoom and whether the order it travels in makes sense (Field Guide 12).

Evidence: The review record at the end of this guide, kept with the release.

Recommended practice.

The pixels, not the stylesheet, are the evidence.

APPENDIX A

The tokens and the rule that draws them.

Excerpts from the kit. The focus files add two tokens to Field Guide 03's semantic layer; all seven files validate against the official DTCTG 2025.10 schemas.

tokens/focus/light.tokens.json (excerpt)

```
{
  "color": { "$type": "color", "border": {
    "focus-inner": { "$value": "{core.color.neutral.0}",
      "$description": "Inner band of the two-tone focus ring, between the component and
color.border.focus." } } },
  "stroke": { "$type": "dimension",
    "focus-inset": { "$value": "{core.stroke.300}",
      "$description": "Focus ring width when drawn inside the component. 3px: a 2px inset ring
is smaller than the WCAG 2.4.13 minimum area." } }
}
```

Merge into Field Guide 03's semantic files for production. Its rules checker accepts hyphenated stroke names such as `stroke.focus-inset` from Field Guide 03 version 1.1 onward; verified by running it on the merged files.

styles/focus-ring.css (excerpt)

```
:focus-visible {
  outline: var(--ds-stroke-focus) solid var(--ds-color-border-focus);
  outline-offset: var(--ds-stroke-focus);
  box-shadow: 0 0 0 var(--ds-stroke-focus) var(--ds-color-border-focus-inner);
}
[role="menuitem"]:focus-visible, [role="option"]:focus-visible,
[role="gridcell"]:focus-visible, .ds-focus-inset:focus-visible {
  outline-width: var(--ds-stroke-focus-inset);
  outline-offset: calc(-1 * var(--ds-stroke-focus-inset));
  box-shadow: none;
}
[data-surface="brand"] {
  --ds-color-border-focus: var(--ds-color-text-on-brand);
  --ds-color-border-focus-inner: var(--ds-color-bg-brand);
}
@media (forced-colors: active) {
  :focus-visible { outline-color: Highlight; }
}
```

The universal `box-shadow` replaces any elevation shadow on the focused element; give elevated components their own `:focus-visible` rule that lists both shadows.

APPENDIX B

Six common rings, measured.

`check-focus-ring.mjs --ring rings/violations.ring.json` over the same 7 contexts in both themes: 56 checks per design. Each design ships in real products.

Design	Fails	What fails, and why
Kit two-tone ring	0	Nothing: 56 of 56 pass.
1px ring, 1px gap	32	2.4.13 and forced colors in all 14 contexts: half the required area. Invisible on the brand banner.
One colour for both themes (#80adff)	24	2.25:1 on white: 1.4.11, 2.4.13 and separation in every light context, plus the dark banner.
Box-shadow ring, <code>outline: none</code>	24	Forced colors in all 14 contexts; separation on the five filled or bordered controls in each theme.
Single 2px ring, no offset	10	Separation only: passes WCAG, reads as a bigger button.
Kit ring without the brand swap	2	Separation on the trial banner, light and dark.
2px ring, 2px gap (the essay's ring)	4	Only where it must be drawn inside: 928 of 944 px ² on the menu item.

96 failures across the six designs. The essay's `outline: 2px` with `outline-offset: 2px` holds up everywhere it can be drawn outside the component.

terminal

```
$ node scripts/measure-focus.mjs examples/focus-demo.violations.html
2 [light] button "Invite teammate"      1112 / 730 px2  ok
5 [light] input "Email address"          704 / 1419 px2  FAIL 2.4.13: 3:1 area too small
8 [light] button "Remove from team"      0 / 944 px2    FAIL 2.4.7: nothing changes
9 [light] a "Pending invitation: ana@example.com"
                                         0 / 1408 px2  FAIL 2.4.7: nothing changes
11 [dark] button "Invite teammate"       0 / 730 px2    FAIL 1.4.11: no pixel changes by 3:1
18 stops: 7 meet 2.4.13, 3 too small, 4 below 3:1, 4 with no visible change
```

Excerpt. Stop 5 is Chromium's default ring, left alone by the author; stop 11 is a ring hard-coded for light mode. The kit demo measures 18 of 18.

KEEP WITH THE RELEASE

Leave a focus ring review record.

One record per release that touches tokens, surfaces or focus CSS.

Release / token package version	Focus tokens changed (and why)
Surfaces added to focus-contexts.json	Ring checker run (link)
Focus CSS lint on compiled CSS (link)	Pixel measurement, normal and forced colors (link)
Windows contrast theme check (build, theme, date)	200% zoom check of key routes
Exceptions and their expiry	Owner / next review

A lighter ring is a token change, not a component change.

If someone asks for a subtler ring on one component, the answer is a proposal against R01 to R03 with the checker output attached.

SOURCES / MAINTENANCE

Keep the guide current.

Sources checked 24 September 2026. Every number in the guide was produced by running the kit on that date with the versions on the cover.

S01 / W3C, WCAG 2.2

W3C Recommendation, 12 December 2024. Success criteria and levels.
<https://www.w3.org/TR/WCAG22/>

S02 / W3C, Understanding SC 2.4.7 Focus Visible

AA; sets no contrast or size, defers to 1.4.11 and 2.4.13.
<https://www.w3.org/WAI/WCAG22/Understanding/focus-visible.html>

S03 / W3C, Understanding SC 1.4.11 Non-text Contrast

Focus indicators need 3:1 against adjacent colours; inside, outside and partial indicators.
<https://www.w3.org/WAI/WCAG22/Understanding/non-text-contrast.html>

S04 / W3C, Understanding SC 2.4.11 Focus Not Obscured (Minimum)

AA; sticky headers; scroll-padding as a sufficient technique.
<https://www.w3.org/WAI/WCAG22/Understanding/focus-not-obscured-minimum.html>

S05 / W3C, Understanding SC 2.4.13 Focus Appearance

AAA; minimum area formulas, the 90 by 30 example, two-colour and inset indicators.
<https://www.w3.org/WAI/WCAG22/Understanding/focus-appearance.html>

S06 / W3C, Technique C40: two-colour focus indicator

Sufficient for 1.4.11, 2.4.7 and 2.4.13; two bands of at least 2px.
<https://www.w3.org/WAI/WCAG22/Techniques/css/C40>

S07 / Sara Soueidan, designing accessible focus indicators

Two-tone rings, outline-offset, outlines that survive forced colors (updated 2023).
<https://www.sarasoueidan.com/blog/focus-indicators/>

S08 / GOV.UK Design System, focus states

Yellow and black two-colour focus styles for links and inputs.
<https://design-system.service.gov.uk/get-started/focus-states/>

S09 / MDN, :focus-visible

When browsers show focus; Baseline widely available since March 2022.
<https://developer.mozilla.org/en-US/docs/Web/CSS/:focus-visible>

S10 / MDN, forced-colors media feature

box-shadow forced to none; outline-color among the overridden properties.
<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/forced-colors>

S11 / W3C, CSS Color Adjustment Module Level 1

Candidate Recommendation Snapshot, December 2025. Authored system colours are kept in forced colors mode.
<https://www.w3.org/TR/css-color-adjust-1/>

S12 / DTCG, Format module 2025.10

\$value, \$type, \$description, dimension objects and aliases.
<https://www.designtokens.org/TR/2025.10/format/>

S13 / Playwright, page.emulateMedia

forcedColors: 'active' or 'none' for testing forced colors mode.
<https://playwright.dev/docs/api/class-page#page-emulate-media>

Maintenance: re-run `npm run check` after any change to surfaces, themes or focus CSS, and after each Playwright upgrade (screenshots and forced colors emulation are Chromium features). Update the WCAG references if WCAG 3 reaches Recommendation. Update the PDF, HTML, Markdown and JSON together.