
A RESOURCE FOR DESIGN SYSTEM AND ENGINEERING TEAMS

Shadow DOM token scoping rules.

Tokens flow in. Styles stay put.

Read the theme. Never redefine it.

24

rules, each with its check

12

checks the linter runs on CSS
and Lit

16

boundary probes you can run in
any browser

The boundary is a filter, not a wall.

A shadow root stops selectors, not inheritance. Every custom property set on the document reaches every element inside every shadow root, and so do inherited properties such as `color` and `font-family` [S01, S02]. Class selectors, element selectors, utility classes and global resets do not. That is exactly the split a token system wants: the theme travels in as custom properties, and the component's own styles cannot leak out.

The split only holds if components follow a few rules. A component that redefines `--ds-color-bg-brand` on `:host` silently re-themes everything slotted into it. A literal fallback in `var()` hides a missing token sheet. `:host-context()` works in Chromium and nowhere else. A registered property with `inherits: false` stops at the first boundary. Each of these ships without an error, and each is a one-line check.

This guide gives 24 rules. The kit's checker runs 12 checks covering 15 of them, on plain CSS and on Lit `css` templates, reporting 17 errors and 1 warning on the failing example and none on the clean one or on the Field Guide 15 starter. The probe page measures the rest in whatever browser you open it in.

Version 1.0 / Sources checked 24 September 2026

Field Guide 14 of the Design × AI series; pairs with 13 and 15. Probe results measured 24 September 2026 in Chromium 153 (Playwright 1.63). Token names follow Field Guide 03.

Practical guidance, not a standard. The checker is a focused heuristic, not a CSS parser; it reads the rules listed in Appendix A and nothing else. Firefox and Safari results come from MDN and browser-compat-data, not from the probe run. Prepared with AI assistance and edited by hand.

START HERE

What crosses the boundary.

Measured by the kit's `boundary-probe.html` in Chromium 153. Open the file in each browser you support; every row sets up a real shadow root and reports what it saw.

Crosses into the shadow root	Stops at the boundary
Custom properties set on <code>:root</code> or any ancestor (P01)	Class and element selectors from the page (P03)
Inherited properties: <code>color</code> , <code>font-*</code> , <code>line-height</code> (P02, P16)	Global resets: <code>*</code> , <code>html</code> , <code>body</code> rules (S01)
<code>color-scheme</code> , so <code>light-dark()</code> resolves per theme (P15)	Registered properties with <code>inherits: false</code> (P06)
<code>@font-face</code> names declared in the document (S01)	The page's <code>@layer</code> order (P13)
Page rules on the host, which beat <code>:host</code> (P08)	<code>@property</code> rules declared inside the shadow root (P07)

Row ids are the probe's. `all: unset` inside the shadow root leaves custom properties in place (P05), so a component reset never deletes the theme.

Building a new component

Sections 01 and 02, then copy the clean example's structure: three layers, private `--_` state on `:host`, semantic tokens only.

Auditing an existing library

Run `check-shadow-css.mjs` over the components folder. It reads `.css` files and Lit `css` templates in `.ts` and `.js`.

Opening a styling API

Section 03: parts for layout tweaks, `--<tag>-*` properties for documented hooks, tokens for everything a theme changes.

Server rendering

Rule T22. Declarative shadow DOM inlines styles per instance; measure the cost before you promise SSR.

Label	Meaning
CHECKER	Enforced by <code>check-shadow-css.mjs</code> ; the rule id (SD01 to SD12) is in the evidence line.
PROBE	Measured by a row (P01 to P16) of <code>boundary-probe.html</code> .
KIT TEST	Observed by the Field Guide 15 starter's browser tests.
SPEC	Follows from a CSS specification or MDN, with no automated check in the kit.
PRACTICE	A working method with a review signal rather than a hard check.

SECTION 01

Tokens in: read the theme, never set it.

The document owns token values. Components read them through inheritance and keep their own state in private properties.

Suggested owners: Design-system lead + component owner

☐ Load token values once, in the document

T01 **PROBE** **CHECKER**

Custom properties inherit through every shadow boundary, so one token stylesheet on the page reaches every component. Importing tokens inside each shadow root duplicates them and breaks theme switching.

Evidence: P01 reads the `:root` value inside a shadow root. SD10 flags `@import` in component CSS.

CSS Shadow: the flat tree is used for inheritance (S01); Lit styles docs (S02).

☐ Components read semantic tokens only

T02 **CHECKER**

`--ds-color-bg-brand`, never `--ds-core-color-blue-600`. A component may read its own component tokens (`--ds-comp-button-*`) and no one else's. The layering is Field Guide 03's, unchanged by the shadow root.

Evidence: SD04 flags `--ds-core-*` and other components' `--ds-comp-*`.

Field Guide 03, rules R11 and R13.

☐ A component never sets a global token

T03 **CHECKER** **SPEC**

Slotted elements inherit from their slot, not from their light-DOM parent. A button that sets `--ds-color-text-default` on `:host` re-themes the consumer's own markup slotted into it.

Evidence: SD05 flags any `--ds-*` declaration that is not the component's documented hook.

CSS Shadow: elements assigned to a slot inherit from that slot (S01).

☐ Private state lives in `--_` properties on `:host`

T04 **CHECKER**

Variants switch private properties (`--_bg`, `--_fg`); the inner elements read only those. One place per variant, and no name that can collide with the page.

Evidence: SD06 flags a custom property that is neither `--_*` nor `--<tag>-*`.

Lea Verou, pseudo-private custom properties (S14).

☐ No raw colours or lengths

T05 **CHECKER**

No hex, no colour functions, no named colours, no `px` or `rem` in spacing, sizing, radius, border, outline or font size. `transparent`, `currentColor`, `0`, `em` and percentages stay allowed.

Evidence: SD01 and SD02. The clean example and the Field Guide 15 styles pass with zero findings.

Field Guide 03, rule R15, applied inside the shadow root.

☐ **System colours only in forced-colors mode**T06 **CHECKER**

`ButtonText`, `Canvas` and the other system colours belong inside `@media (forced-colors: active)`. Elsewhere they bypass the theme.

Evidence: SD01 allows system colours only inside the forced-colors block.

Kit rule.

☐ **No literal fallbacks in `var()`**T07 **CHECKER**

`var(--ds-radius-control, 6px)` copies today's value into the component and hides a missing token sheet. Fall back to another token or not at all, so a broken theme fails loudly in review.

Evidence: SD03 flags a fallback that is not itself a `var()`.

Recommended practice.

☐ **Do not register tokens with `inherits: false`**T08 **PROBE** **CHECKER**

A registered property that does not inherit stops at every boundary, and browsers honour `@property` only from the document. Register tokens in the document with `inherits: true`, or not at all.

Evidence: P06 shows `0px` (the initial value) inside the shadow root; P07 shows a shadow-root `@property` ignored. SD10 flags `@property` in component CSS.

MDN `@property`; the spec says registrations are global, but csswg issue 10541 records that browsers disagree (S05, S06, S07).

☐ **Reset with `all`, knowing tokens survive it**T09 **PROBE**

`all: unset` on the inner control removes browser button styles and leaves every custom property in place, because `all` does not reset custom properties.

Evidence: P05: the token is still readable after `all: unset`.

MDN `all` (S04).

The page owns the theme. The component only reads it.

SECTION 02

Selectors stay inside the shadow root.

Selectors never cross a boundary in either direction. Rules that assume they do are dead code at best.

Suggested owners: Component owner

☐ No global selectors in component CSS

T10 **CHECKER** **PROBE**

`:root`, `html`, `body` and a bare `*` match nothing useful inside a shadow root. Style the host with `:host` and the internals by their own selectors.

Evidence: SD07. P04: a `:root` rule inside the shadow root matches nothing.

Selectors are limited to a single tree (S01).

☐ Theme by tokens on an ancestor, not `:host-context()`

T11 **CHECKER** **KIT TEST**

`:host-context()` is Chromium-only: Firefox and Safari do not support it, and MDN marks it deprecated. Switch the theme by changing token values on an ancestor such as `[data-theme]`.

Evidence: SD08. Field Guide 15 test "follows the theme without re-rendering" flips `data-theme` and reads the new brand colour.

MDN `:host-context` and `browser-compat-data` (S09).

☐ Style slotted content lightly, with compound selectors

T12 **CHECKER** **PROBE**

`::slotted()` takes one compound selector and matches only top-level slotted elements; slotted nodes belong to the page, whose rules win. Use it for spacing and icon size, not for typography the consumer controls.

Evidence: SD11. P12: `::slotted(span b)` matches nothing.

MDN `::slotted`; CSS Cascade 5 context (S12, S08).

☐ Ship your own minimal reset

T13 **PROBE**

The page's reset, utility classes and component classes never reach inside. Put a small reset in `@layer reset` in each component, not a copy of the page's.

Evidence: P03: a page class selector does not match an inner element with the same class.

S01, S02, S03.

☐ Fonts are declared in the document

T14 **CHECKER** **PROBE**

`@font-face` inside a shadow root is unreliable (a long-standing Chromium issue). Declare fonts in the page; `font-family` then inherits into every component, or read it from a token.

Evidence: SD10 flags `@font-face` in component CSS. P16: a font family set on `body` reaches shadow text.

S01, S17.

What the page cannot select, it cannot break.

SECTION 03

Hooks out: a small styling API.

Give consumers three levels, in this order: tokens for themes, `--<tag>-*` properties for documented hooks, parts for layout tweaks.

Suggested owners: Design-system lead + component owner

☐ `:host` styles are defaults the page can override

T15 **CHECKER** **PROBE**

For normal declarations the outer context wins, whatever the specificity. `!important` inside the shadow root reverses that and takes the override away.

Evidence: P08: page beats `:host`. P09: `:host` `!important` beats the page. SD09.

CSS Cascade 5, context (S08).

☐ Expose parts by role, and document them

T16 **SPEC** **KIT TEST**

`part="control"` on the native button, `@csspart control` in the JSDoc, so the manifest and Storybook list it.

Evidence: The Field Guide 15 manifest lists one part, `control`. P11: the page styles it with `::part(control)`.

MDN `::part` (S10).

☐ Forward nested parts with `exportparts`

T17 **SPEC**

`::part()` cannot be chained, so a nested component's part is unreachable unless the outer one forwards it: `exportparts="control: button-control"`.

Evidence: Every nested part a consumer needs is in the outer `exportparts` and manifest.

MDN `::part`, `exportparts` (S10, S11).

☐ Public hooks are `--<tag>-*`, default to a token

T18 **CHECKER** **PRACTICE**

`var(--ds-button-radius, var(--ds-radius-control))`: the name says who owns it, `@cssprop` documents it, the default keeps the theme in charge.

Evidence: SD06 allows `--ds-button-*` only in `ds-button`; SD03 allows the token fallback.

Web Awesome scopes component properties per component (S15).

☐ A theme changes tokens, not parts

T19 **PRACTICE**

Parts are per-component escape hatches. A theme that reaches for `::part()` across components has become a second design system.

Evidence: Theme files contain token declarations only.

CSS Shadow on `::part()` (S01); Shoelace token guidance (S16).

Three levels of override, and each one is written down.

SECTION 04

Cascade and delivery.

Ordering inside the component, sharing between instances, sending from the server.

Suggested owners: Engineering lead + component owner

☐ Order component CSS with `@layer`

T20 **PROBE**

`@layer reset, component, state;` keeps state styles winning over variants without specificity games. The page's layer order does not apply inside.

Evidence: P13: the page's `@layer b, a` order does not reach inside.

CSS Cascade 5: layers are scoped to their context (S08).

☐ Share one stylesheet across instances

T21 **PROBE** **SPEC**

A constructed stylesheet adopted by many shadow roots is parsed once and updates everywhere. Lit's `static styles` does this.

Evidence: P14: one `replaceSync` updates both adopting roots.

MDN `adoptedStyleSheets`, Baseline 2023; Lit styles (S13, S02).

☐ Budget for style duplication in server rendering

T22 **KIT TEST**

Declarative shadow DOM repeats the styles in every server-rendered instance: 2,671 of 3,087 bytes per `ds-button`. Declarative sharing is only in a Chrome origin trial.

Evidence: `npm run ssr` in the Field Guide 15 starter.

Kit run; blink-dev intent to experiment (S19).

☐ Set `color-scheme` with the theme

T23 **PROBE**

`color-scheme` inherits into shadow roots, so `light-dark()` and native controls inside follow the theme.

Evidence: P15: `light-dark()` picks the dark value when the host has `color-scheme: dark`.

MDN `light-dark()` (S18).

☐ Every animation has a reduced-motion answer

T24 **CHECKER**

Spinners and transitions live inside the component, so the reduced-motion override must too.

Evidence: SD12 warns when a stylesheet animates without a `prefers-reduced-motion` block.

Kit rule; WCAG 2.2 SC 2.3.3 (AAA) intent.

Parse once, share everywhere, and measure what the server sends.

APPENDIX A

Twelve checks, and what they caught.

`node scripts/check-shadow-css.mjs <paths>` reads `.css` files and every Lit `css` template in `.ts` and `.js` files. Exit code 1 on any error.

Check	Flags	Rule
SD01	Raw colours; system colours outside forced-colors	T05, T06
SD02	<code>px</code> or <code>rem</code> in spacing, sizing, radius, border, outline, font size	T05
SD03	<code>var()</code> with a literal fallback	T07
SD04	<code>--ds-core-*</code> or another component's <code>--ds-comp-*</code>	T02
SD05	A global <code>--ds-*</code> token declared inside the component	T03
SD06	A custom property that is neither <code>--_*</code> nor <code>--<tag>-*</code>	T04, T18
SD07	<code>:root</code> , <code>html</code> , <code>body</code> or a bare <code>*</code>	T10
SD08	<code>:host-context()</code>	T11
SD09	<code>!important</code>	T15
SD10	<code>@import</code> , <code>@font-face</code> or <code>@property</code>	T01, T08, T14
SD11	<code>::slotted()</code> with a complex selector	T12
SD12	Animation or transition without reduced motion (warning)	T24

```
node scripts/check-shadow-css.mjs examples/violations
```

```
ds-button.css:6    error SD10 @import inside a shadow root
ds-button.css:8    error SD10 @font-face inside a shadow root
ds-button.css:13   error SD10 @property inside a shadow root
ds-button.css:19   error SD07 global selector ":root" matches nothing useful
ds-button.css:20   error SD01 --ds-color-bg-brand: #125bd0 uses a raw colour
ds-button.css:20   error SD05 --ds-color-bg-brand redefines a global token inside the component
ds-button.css:23   error SD07 global selector "*" matches nothing useful
ds-button.css:28   error SD09 display: inline-flex !important takes the override from the page
ds-button.css:29   error SD06 --bg is private state; prefix it --_
ds-button.css:32   error SD08 :host-context() is unsupported in Firefox and Safari
ds-button.css:33   error SD04 --ds-core-color-blue-300 is not a semantic token
ds-button.css:37   error SD02 padding: 8px 16px hard-codes a length
ds-button.css:38   error SD03 var(--ds-radius-control, 6px) hard-codes a fallback
ds-button.css:40   error SD01 color: white uses a raw colour
ds-button.css:41   error SD01 outline-color: rgb(18 91 208) uses a raw colour
ds-button.css:42   warn  SD12 animation or transition with no prefers-reduced-motion block
ds-button.css:45   error SD11 ::slotted(span svg) is complex; only compound selectors match
ds-button.css:46   error SD01 fill: ButtonText uses a system colour outside forced-colors
shadow css: 1 file(s), 17 error(s), 1 warning(s)
```

Real output, with the directory prefix removed and messages shortened to fit the page. The same command over `examples/clean` and the Field Guide 15 `src/ds-button.styles.ts` prints `2 file(s), 0 error(s), 0 warning(s)`.

APPENDIX B

The clean button, and what the page owns.

Excerpts from `examples/clean/ds-button.css` and `examples/document.css`. The component reads; the document declares.

`examples/clean/ds-button.css` (excerpt)

```
@layer reset, component, state;

@layer component {
  :host {
    --_bg: var(--ds-color-bg-brand);
    --_fg: var(--ds-color-text-on-brand);
    --_pad: var(--ds-space-inline-md);
  }
  :host([tone="destructive"]) {
    --_bg: var(--ds-color-bg-danger);
    --_fg: var(--ds-color-text-on-danger);
  }
  [part="control"] {
    padding-inline: var(--_pad);
    border-radius: var(--ds-button-radius, var(--ds-radius-control));
    background: var(--_bg);
    color: var(--_fg);
  }
}

@media (forced-colors: active) {
  [part="control"] { border-color: ButtonText; }
}
```

`examples/document.css` (condensed)

```
@font-face { font-family: "Brand Sans"; src: url("/fonts/brand-sans.woff2") format("woff2"); }

:root { color-scheme: light; --ds-color-bg-brand: #125bd0; }
[data-theme="dark"] { color-scheme: dark; --ds-color-bg-brand: #80adff; }

ds-button.hero { --ds-button-radius: 999px; } /* documented hook */
ds-button::part(control) { letter-spacing: 0.01em; } /* local tweak */
```

Raw values are right here: this is the generated token layer, not component CSS. The checker is for components; run the Field Guide 03 gate on the document side.

APPENDIX C

Sixteen probes in Chromium 153.

`boundary-probe.html` with no server and no dependencies. Open it in Firefox and Safari too; the only row expected to differ is P10.

Id	Question	Observed
P01	Custom property from <code>:root</code> inside a shadow root	<code>rgb(18, 91, 208)</code> : crosses
P02	Inherited <code>color</code> from the host	crosses
P03	Page class selector matches an inner element	no
P04	<code>:root</code> rule inside the shadow root matches	no
P05	<code>all: unset</code> removes inherited custom properties	no
P06	<code>inherits: false</code> property from the host	<code>0px</code> (initial value)
P07	<code>@property</code> declared inside a shadow root honoured	no
P08	Page rule vs <code>:host</code> rule	page wins
P09	Page rule vs <code>:host !important</code>	<code>:host</code> wins
P10	<code>:host-context()</code> supported	yes (Chromium only)
P11	<code>::part(label)</code> styled by the page	yes
P12	<code>::slotted(span b)</code> matches	no
P13	Page <code>@layer</code> order applies inside	no
P14	One adopted sheet updates two roots	2 of 2
P15	<code>light-dark()</code> follows the host's <code>color-scheme</code>	yes
P16	<code>font-family</code> on <code>body</code> reaches shadow text	yes

All 15 expectation rows matched in Chromium 153; P10 is informational. Firefox and Safari were not run for this guide; their P10 answer (no) comes from browser-compat-data via MDN (S09).

KEEP WITH THE COMPONENT

Leave a styling review record.

One record per component release. It shows which parts of the styling API are public and what was checked.

Component / version	Checker run: errors and warnings (link)
Tokens read (semantic list)	Public hooks (--tag-*) and defaults
Parts exposed / exportparts	Probe run per browser (links)
Forced-colors and reduced-motion review	SSR payload per instance (bytes)
Waived checks and reasons	Owner / next review

A new public hook is an API change.

Adding `--ds-button-*` or a part is a minor release; renaming or removing one is major, exactly like a semantic token (Field Guide 03, R23).

SOURCES / MAINTENANCE

Keep the guide current.

Sources checked 24 September 2026. Baseline dates come from web-features via MDN. Behaviour marked PROBE was measured in Chromium 153 only.

S01 / CSS Shadow Module Level 1 (formerly CSS Scoping)

Flat-tree inheritance, slot inheritance, featureless host, `::part` and tree-scoped names.

<https://drafts.csswg.org/css-shadow-1/>

S02 / Lit, styles

Custom properties and inherited properties cross the boundary; `:host` defaults; adopted stylesheets.

<https://lit.dev/docs/components/styles/>

S03 / MDN, Using shadow DOM

Page CSS does not affect nodes inside a shadow root.

https://developer.mozilla.org/en-US/docs/Web/API/Web_components/Using_shadow_DOM

S04 / MDN, all

Resets every property except unicode-bidi, direction and custom properties.

<https://developer.mozilla.org/en-US/docs/Web/CSS/all>

S05 / MDN, @property

inherits: false; Baseline July 2024.

<https://developer.mozilla.org/en-US/docs/Web/CSS/@property>

S06 / CSS Properties and Values API Level 1

Registrations are global by spec, not scoped to a tree.

<https://drafts.css-houdini.org/css-properties-values-api/>

S07 / csswg-drafts issue 10541

Implementations honour `@property` from the document scope only; open.

<https://github.com/w3c/csswg-drafts/issues/10541>

S08 / CSS Cascade Level 5

Context sorts before layers and specificity; layers are scoped per tree.

<https://drafts.csswg.org/css-cascade-5/>

S09 / MDN, :host-context()

Deprecated; not supported in Firefox or Safari.

<https://developer.mozilla.org/en-US/docs/Web/CSS/:host-context>

S10 / MDN, ::part()

Pseudo-classes allowed after it, structural ones not; cannot be chained.

<https://developer.mozilla.org/en-US/docs/Web/CSS::part>

S11 / MDN, exportparts

Forwards and renames parts from nested shadow trees.

https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes/exportparts

S12 / MDN, ::slotted()

Compound selectors only; elements, not text nodes.

<https://developer.mozilla.org/en-US/docs/Web/CSS::slotted>

S13 / MDN, Document.adoptedStyleSheets

One constructed sheet shared by many roots; Baseline March 2023.

<https://developer.mozilla.org/en-US/docs/Web/API/Document/adoptedStyleSheets>

S14 / Lea Verou, Custom properties with defaults: 3+1 strategies

Pseudo-private `--_` properties (2021).

<https://lea.verou.me/blog/2021/10/custom-properties-with-defaults/>

S15 / Web Awesome, customizing

Component-scoped custom properties versus `--wa-` theme tokens; parts.

<https://webawesome.com/docs/customizing>

S16 / Shoelace, customizing

Design tokens live on `:root` with a library prefix.

<https://shoelace.style/getting-started/customizing>

S17 / Chromium issue 41085401

@font-face definitions in a shadow root cannot be used within it.

<https://issues.chromium.org/issues/41085401>

S18 / MDN, light-dark()

Resolves against the used color scheme; Baseline 2024.

<https://developer.mozilla.org/en-US/docs/Web/CSS/light-dark>

S19 / blink-dev, Intent to Experiment: declarative CSS module scripts

shadowrootadoptedstylesheets in a Chrome origin trial (March 2026).

[http://www.mail-archive.com/blink-dev@chromium.org/msg16211.h](http://www.mail-archive.com/blink-dev@chromium.org/msg16211.html)
tml

Maintenance: re-run the probe in every supported browser at each major release and update Appendix C. Recheck csswg issue 10541 (@property scoping) and the declarative stylesheet origin trial; if either lands, rules T08 and T22 change. Update the PDF, HTML, Markdown and JSON together.