Nexus

Extension Style Guide

Design tokens, CSS classes, and components for building native-feeling extensions

nexusprism.org · MIT License
Covers Nexus v0.3.8-beta and later

About this guide

This guide documents everything an extension author needs to make their extension look and feel like part of Nexus — the design tokens, reusable CSS classes, React components, and layout patterns Nexus itself uses.

All values here come directly from Nexus's source code. Following this guide means your extension will automatically adapt to whatever accent color, theme, or mode the admin has configured.

■ ***Never hardcode hex color values for anything that should respond to the admin's color scheme. Use CSS variables exclusively for colors, surfaces, borders, and text.***

1. Design tokens — CSS variables

All Nexus design decisions are expressed as CSS custom properties on `:root`. These update automatically when the user switches between dark and light mode, or when the admin changes the accent or tint color. Always use these rather than fixed values.

1.1 Backgrounds and surfaces

Variable	Dark default	Light default	Use for
`--bg`	`#111111`	`#f4f4f5`	Page background — outermost layer
`--s1`	`#1a1a1a`	`#ffffff`	Cards, panels, content blocks
`--s2`	`#222222`	`#e4e4e7`	Input fields, nested cards, dropdowns
`--s3`	`#2a2a2a`	`#d4d4d8`	Tooltips, popovers, overlays

1.2 Borders

Variable	Dark default	Use for
`--b1`	`rgba(255,255,255,0.07)`	Subtle dividers, section separators
`--b2`	`rgba(255,255,255,0.10)`	Standard borders — cards, inputs
`--b3`	`rgba(255,255,255,0.14)`	Emphasis borders — focused states

■ Border variables use rgba with opacity, not solid colors, so they adapt correctly over any background.

1.3 Text

Variable	Dark default	Use for
`--t1`	`#f0eeff`	Primary — headings, titles, important labels
`--t2`	`rgba(255,255,255,0.75)`	Secondary — body copy, descriptions
`--t3`	`rgba(255,255,255,0.55)`	Tertiary — captions, helper text
`--t4`	`rgba(255,255,255,0.38)`	Muted — placeholders, disabled labels
`--t5`	`rgba(255,255,255,0.28)`	Faintest — barely-visible hints

1.4 Accent

Accent variables are set by the admin from **Admin** → **Appearance**. Never override these — they are the forum's identity color and the admin's explicit choice.

Variable	Use for
`--ac`	Accent color — primary buttons, active states, selected items
`--ac-on`	Text drawn directly on top of the accent color
`--ac-bg`	Faint accent-tinted background — selected state, active tab fills
`--ac-border`	Accent-tinted border — focused inputs, selection rings
`--ac-text`	Accent-tinted text on neutral backgrounds — active labels, links

✓ For interactive elements, use --ac for the fill and --ac-text for text labels. For selected states, use --ac-bg as the background and --ac-border as the border.

1.5 Semantic colors

Variable	Dark default	Use for
`--green`	`#34d399`	Success, positive values, online indicators
`--red`	`#f87171`	Errors, destructive actions, danger
`--amber`	`#fbbf24`	Warnings, pending states
`--blue`	`#60a5fa`	Informational states
`--pink`	`#f472b6`	Decorative accents

1.6 Typography

Variable	Default	Controls
`--fs-ui`	`16px`	UI chrome — navigation, sidebar labels, admin text
`--fs-body`	`13px`	General body text — descriptions, card content
`--fs-title`	`20px`	Page-level titles
`--fs-feed-title`	`14px`	Post titles in the feed list
`--fs-content`	`14px`	Post body content area
`--fs-code`	`12px`	Code blocks and inline code

The base font is **Inter** (loaded via Google Fonts). All Nexus UI uses `font-family: 'Inter', system-ui, sans-serif`. Use `font-size: var(--fs-body)` as your default text size.

1.7 Shape

Variable	Default	Controls
`--av-radius`	`22%`	Avatar border-radius. 0% = square, 50% = circle. Applies to all user avatars across the forum.
`--tgl-off`	rgba	Toggle background in off state. Do not use directly — use the `.tgl` class instead.

2. Reusable CSS classes

These classes are defined in Nexus's stylesheet and available in every page. Use them in your extension's JSX to match Nexus's visual language without duplicating styles.

2.1 Buttons

Class	Description	When to use
`btn-primary`	Filled, accent-colored button. 14px, rounded pill.	Primary actions — Submit, Save, Install
`btn-ghost`	Transparent with subtle border. 13px, rounded pill.	Secondary actions — Cancel, View, Less important actions
`write-btn`	Larger accent button with heavier weight. 15px.	Compose / create actions — used for the main compose button
`icon-btn`	40×40px circular icon button with subtle background.	Toolbar icons, action icons, small controls

```
&lt;button className="btn-primary"&gt;Save settings&lt;/button&gt;
&lt;button className="btn-ghost"&gt;Cancel&lt;/button&gt;
&lt;button className="icon-btn"&gt;
  &lt;i className="fa-solid fa-ellipsis" /&gt;
&lt;/button&gt;
```

2.2 Form elements

Class	Description
`fg`	Form group wrapper — applies consistent bottom margin (18px) between fields
`fl`	Form label — 13px, `--t3` color, block display
`fi`	Form input — full-width, rounded (12px), subtle background, accent focus ring. Use on `<input>`, `<textarea>`, `<select>`.
`f-label`	Alternative label — 14px, `--t3` color. For use outside auth forms.
`f-hint`	Helper text below a field — 12px, `--t5` color
`ferr`	Validation error message — 13px, `--red` color
`fgt`	Section group title — uppercase, small, border-bottom. Use to separate form sections.

```
&lt;div className="fg"&gt;
  &lt;label className="fl"&gt;API key&lt;/label&gt;
  &lt;input className="fi" type="password" placeholder="sk-..." /&gt;
  &lt;div className="f-hint"&gt;Your API key from the dashboard&lt;/div&gt;
&lt;/div&gt;

&lt;div className="fg"&gt;
  &lt;label className="fl"&gt;Webhook URL&lt;/label&gt;
  &lt;input className="fi" type="text" /&gt;
  &lt;div className="ferr"&gt;Please enter a valid URL&lt;/div&gt;
&lt;/div&gt;

&lt;div className="fgt"&gt;Advanced&lt;/div&gt;
```

2.3 Panels and layout

Class	Description
`panel`	Content panel — subtle background, border, 14px border-radius, 20px padding. The standard container for content blocks.
`panel-title`	Panel heading — 14px, `--t3` color, flex between for title + action
`toggle-row`	A label+toggle pair — flex, space-between, bottom border. Use when listing multiple toggles.
`atbl`	Data table — full-width, collapsed borders, uppercase `--t5` headers, `--t3` cell text, subtle hover
`atbl-wrap`	Wraps `atbl` for horizontal scroll on mobile
`link`	Inline link — accent color, pointer cursor
`sp-tag`	Small pill badge — 9px uppercase. Use for status labels, type badges.

```
&lt;div className="panel"&gt;
  &lt;div className="panel-title"&gt;
    &lt;span&gt;Connection settings&lt;/span&gt;
    &lt;button className="btn-ghost"&gt;Test&lt;/button&gt;
  &lt;/div&gt;
  {/* content */}
&lt;/div&gt;

&lt;table className="atbl"&gt;
  &lt;thead&gt;
    &lt;tr&gt;&lt;th&gt;Name&lt;/th&gt;&lt;th&gt;Status&lt;/th&gt;&lt;th&gt;Created&lt;/th&gt;&lt;/tr&gt;
  &lt;/thead&gt;
  &lt;tbody&gt;
    &lt;tr&gt;
      &lt;td&gt;My item&lt;/td&gt;
      &lt;td&gt;&lt;span className="sp-tag" style={{color:"var(--green)",border:"0.5px solid var(--green)
      &lt;td&gt;2026-06-01&lt;/td&gt;
    &lt;/tr&gt;
  &lt;/tbody&gt;
&lt;/table&gt;
```

2.4 Markdown

Class	Description
`md-body`	Applies all Nexus Markdown styles to children — headings, lists, blockquotes, code, tables, spoilers. Also wires the image lightbox. Use this whenever you render user-authored Markdown content.

Use the `Md` React component (see Section 3) which applies `md-body` automatically, rather than setting the class yourself.

3. React components

These components are exposed on `window.NexusComponents`. They use Nexus's exact internal implementations — the same components the core UI uses — so they automatically respond to theme changes, color settings, and mode switching.

```
const { Toggle, Select, Av, Md, toast } = window.NexusComponents;
```

3.1 `Toggle`

A labeled toggle switch using the `.tgl` and `.toggle-row` classes. Renders a label, optional hint, and an animated toggle.

Prop	Type	Required	Description
`value`	boolean	✓	Current on/off state

Prop	Type	Required	Description
`onChange`	fn(bool)	✓	Called with new boolean value on click
`label`	string		Label text shown to the left of the toggle
`hint`	string		Small helper text below the label

```
const [enabled, setEnabled] = useState(false);

&lt;Toggle
  value={enabled}
  onChange={setEnabled}
  label="Enable notifications"
  hint="Send a push notification when someone replies"
/&gt;
```

3.2 `Select`

A styled native select element using the `.fi` class for consistent form appearance.

Prop	Type	Required	Description
`value`	string	✓	Currently selected value
`onChange`	fn(str)	✓	Called with new value string on change
`options`	array		Array of `{value, label}` objects — alternative to `children`
`children`	JSX		Native `<option>` elements — alternative to `options`
`disabled`	boolean		Disables the select

```
// With options array
&lt;Select
  value={level}
  onChange={setLevel}
  options={[
    {value: "low",    label: "Low"},
    {value: "medium", label: "Medium"},
    {value: "high",   label: "High"},
  ]}
/&gt;

// With native options
&lt;Select value={level} onChange={setLevel}&gt;
  &lt;option value="low"&gt;Low&lt;/option&gt;
  &lt;option value="high"&gt;High&lt;/option&gt;
&lt;/Select&gt;
```

3.3 `Av`

The standard Nexus avatar component. Renders the user's avatar image if they have one, or a generated letter avatar using their username-derived color. Automatically uses `--av-radius` for shape.

Prop	Type	Default	Description
`user`	object	—	User object with at minimum `username` (string). Optional: `avatar_url` (string).
`size`	number	`28`	Width and height in pixels

```
// 28px avatar (default)
&lt;Av user={currentUser} /&gt;

// 40px avatar
&lt;Av user={currentUser} size={40} /&gt;

// In a row with a name
&lt;div style={{display:"flex", alignItems:"center", gap:8}}&gt;
  &lt;Av user={post.user} size={32} /&gt;
  &lt;span style={{color:"var(--t2)"}}&gt;{post.user.username}&lt;/span&gt;
&lt;/div&gt;
```

3.4 `Md`

Renders a Nexus Markdown string as HTML with full styling — headings, lists, code blocks, images, embeds, spoilers, and image grids — with the lightbox wired automatically. Never use this to render untrusted content.

Prop	Type	Required	Description
`text`	string	✓	Markdown string to render

```
&lt;Md text={post.body} /&gt;

// Render a plain description
&lt;Md text="**Bold** and _italic_ text with `code` support." /&gt;
```

3.5 `toast`

Fire-and-forget toast notification. Appears in the bottom-left corner of the screen and auto-dismisses after 4 seconds.

Argument	Type	Description
`msg`	string	The message to display
`type`	string	`"ok"` (green, default), `"err"` (red), or `"warn"` (amber)

```
toast("Settings saved!");         // green success
toast("Failed to connect", "err"); // red error
```

```
toast("Token expires soon", "warn"); // amber warning
```

4. Layout patterns

Extension UI appears inside Nexus's right sidebar, admin panel, profile sidebar, post footer, or in a dedicated SPA page. These patterns match how Nexus builds its own UI in each context.

4.1 Right sidebar widget

Right widgets get a fixed-width column (~300px). Keep them simple and vertical — no horizontal scroll, no wide tables. Use `panel` as the outer container.

```
function MyWidget({ currentUser }) {
  return (
    &lt;div className="panel" style={{marginBottom: 12}}&gt;
      &lt;div className="panel-title"&gt;My Widget&lt;/div&gt;
      &lt;div style={{color: "var(--t3)", fontSize: 13}}&gt;
        {/* widget content */}
      &lt;/div&gt;
    &lt;/div&gt;
  );
}
```

4.2 Admin panel

Admin panels are full-width within the admin content area. Use `fgt` for section headings, `fg`+`fl`+`fi` for form fields, and `btn-primary` for the save action. Match the admin's form density — don't add extra padding.

```
function MyAdminPanel() {
  const [cfg, setCfg] = useState({enabled: false, key: ""});

  function save() {
    // save cfg...
    toast("Saved!");
  }

  return (
    &lt;div&gt;
      &lt;div className="fgt"&gt;Connection&lt;/div&gt;
      &lt;Toggle
        label="Enable integration"
        value={cfg.enabled}
        onChange={v =&gt; setCfg(p =&gt; ({...p, enabled: v}))}
      /&gt;
      &lt;div className="fg" style={{marginTop: 16}}&gt;
        &lt;label className="fl"&gt;API key&lt;/label&gt;
        &lt;input
          className="fi"
          type="password"
```

```
          value={cfg.key}
          onChange={e =&gt; setCfg(p =&gt; ({...p, key: e.target.value}))}
          placeholder="sk-..."
        /&gt;
      &lt;/div&gt;
      &lt;button className="btn-primary" onClick={save}&gt;Save&lt;/button&gt;
    &lt;/div&gt;
  );
}
```

4.3 Extension page

Full SPA pages registered via `registerRoute`. Pages have the full viewport minus the left sidebar and topbar. Avoid setting fixed heights — let content scroll. Use `--bg` as the page background and `--s1` for content cards.

```
function MyPage({ currentUser }) {
  return (
    &lt;div style={{
      padding: "24px 28px",
      maxWidth: 860,
      margin: "0 auto",
    }}&gt;
      &lt;h1 style={{
        fontSize: "var(--fs-title)",
        color: "var(--t1)",
        fontWeight: 700,
        marginBottom: 20,
      }}&gt;
        My Page
      &lt;/h1&gt;
      &lt;div className="panel"&gt;
        {/* content */}
      &lt;/div&gt;
    &lt;/div&gt;
  );
}
```

4.4 Post footer slot

Renders below the post body, above the reply thread. Keep it compact — this space is shared with the post's action row. Avoid tall components here.

```
function PostFooter({ post_id }) {
  return (
    &lt;div style={{
      padding: "8px 0",
      borderTop: "0.5px solid var(--b1)",
      marginTop: 8,
    }}&gt;
      {/* e.g. a poll, an attachment summary, a reaction summary */}
    &lt;/div&gt;
```

```
  );
}
```

5. Spacing, sizing, and shape

Nexus uses a consistent spacing rhythm. Match these values rather than choosing arbitrary numbers.

5.1 Spacing

Context	Value	Notes
Page padding	24px top, 28px horizontal	Content area inside a page (`post-content-wrap`)
Panel padding	20px vertical, 22px horizontal	`.panel` class default
Form group gap	18px	`.fg` bottom margin between fields
Label to input gap	6–7px	`.fl` bottom margin
Section heading gap	16px below	`.fgt` margin
Inline gap (text+icon)	8–12px	Gap between an icon and its label
Card inner gap	10–12px	Vertical gap between items inside a panel

5.2 Border-radius

Nexus uses a tiered radius system. Match the right tier to the element type — mismatched radii are immediately noticeable.

Radius	Elements
`50%` — fully round	Icon buttons (`.icon-btn`), notification dots, toggle knob (`.tgl-knob`), row menu buttons
`24px` — large pill	Search bar (`.tb-search`), compose button (`.write-btn`)
`22px` — pill	Primary buttons (`.btn-primary`)
`20px` — soft pill	Ghost buttons (`.btn-ghost`), tag pills (`.sp-tag`, `.sort-pill`, `.rx-pill`), toggles (`.tgl`), reaction trigger
`16px` — rounded	User cards (`.ucard`), popover sheets
`14px` — card	Panels (`.panel`), dropdowns (`.comp-dd`, `.tb-search-drop`), admin stat cards
`12px` — input	Form inputs (`.fi`), reply box, embeds (`.md-embed`), dropdowns, right widgets (`.rw`)

Radius	Elements
`10px` — small card	Stat cards, toast notifications (`.toast`), reaction picker
`8px` — item	Dropdown items, secondary card elements, quote tooltip
`6px` — tight	Composer toolbar buttons, small image buttons
`4px` — subtle	Reply quote button, inline link highlights, post reply button
`var(--av-radius)`	All user avatars — always use this, never a fixed radius

■ ***Never apply a fixed border-radius to user avatars. Always use var(--av-radius) so they respect the admin's avatar shape setting (0% square to 50% circle).***

✓ When building a card-like container, use 14px. When building an input field, use 12px. When building a button, use 20–22px for a pill shape. These three tiers cover most cases.

6. Rules to follow

Always

- Use CSS variables for **all** colors, backgrounds, borders, and text — never hardcode hex values for anything that should respond to admin color settings.
- Use `var(--fs-body)` or `var(--fs-ui)` for font sizes in UI chrome. Match `--t2` or `--t3` for body text.
- Use `border: 0.5px solid var(--b1)` (or `--b2`, `--b3`) for borders — not `1px` and not an arbitrary color.
- Use `window.NexusComponents.Av` for user avatars — never roll your own avatar rendering.
- Use `window.NexusComponents.toast` for one-time feedback messages.
- Test in both dark and light mode before publishing your extension.
- Use `[data-theme="light"]` selectors when any value needs to differ between modes in your `theme.css`.

Never

- **Never** hardcode accent colors (e.g. `#7c3aed`, `#4A90E2`) — use `var(--ac)` and derived accent variables.
- **Never** use `1px solid` borders — Nexus uses `0.5px solid` throughout for a lighter visual weight.
- **Never** use `font-family: sans-serif` without the Inter fallback chain: `'Inter', system-ui, sans-serif`.
- **Never** import your own copy of React — use `window.React` and `window.ReactDOM`.
- **Never** set `overflow: hidden` on a container that wraps a Nexus `Md` component — the lightbox needs access to the viewport.
- **Never** use `z-index` values above 9000 — Nexus's overlays, modals, and toasts use this range.

- **Never** set `position: fixed` in a right sidebar widget — the sidebar scrolls independently.

7. Icons — Font Awesome

Nexus self-hosts Font Awesome 6 Free. The full solid, regular, and brands sets are available. Use the standard class format:

```
&lt;i className="fa-solid fa-check" /&gt;
&lt;i className="fa-regular fa-clock" /&gt;
&lt;i className="fa-brands fa-github" /&gt;
```

In manifest declarations (toolbar buttons, admin panel icons, explore items, profile tabs), use the **short form** without the style prefix: `fa-check`, `fa-clock`. The one exception is `toolbar_buttons` in the manifest — those require the full class including style prefix (`fa-solid fa-check`).

Icon color should always be set via CSS variables, not inline hex values. For toolbar icons and sidebar items, use `color: var(--t3)` at rest and `color: var(--ac)` for active state.

Appendix — Quick reference

I want to...	Use
Primary action button	`<button className="btn-primary">`
Secondary/cancel button	`<button className="btn-ghost">`
Icon-only button	`<button className="icon-btn">`
Text input / textarea / select	`<input className="fi" />`
Form group wrapper	`<div className="fg">`
Form field label	`<label className="fl">`
Helper text	`<div className="f-hint">`
Validation error	`<div className="ferr">`
Section heading	`<div className="fgt">`
Content panel / card	`<div className="panel">`
Data table	`<table className="atbl">`
Inline link	`<span className="link">`
Small status pill	`<span className="sp-tag">`
Toggle switch with label	`<Toggle value={x} onChange={setX} label="..." />`
Dropdown select	`<Select value={x} onChange={setX} options={[...]} />`
User avatar	`<Av user={user} size={32} />`
Render Markdown	`<Md text={content} />`
Success toast	`toast("Message!")`
Error toast	`toast("Failed", "err")`
Warning toast	`toast("Warning", "warn")`
Accent color	`var(--ac)`
Card border-radius	`14px`
Input border-radius	`12px`
Button border-radius	`20–22px` (pill)
Avatar border-radius	`var(--av-radius)` — always
Card background	`var(--s1)`

I want to...	Use
Standard border	`0.5px solid var(--b2)`
Body text	`color: var(--t2)`
Muted / helper text	`color: var(--t3)` or `var(--t4)`