

CSS Custom Properties & var() Test Page

1 — Basic Declaration & Usage

--main-bg, --main-color and --main-border are declared once on the surrounding container and consumed via var() on each box.

Aa

background + color + border
--main-bg: #2c3e50 · --main-color: white · --main-border: 3px solid #1a252f
background: var(--main-bg); color: var(--main-color); border: var(--main-border)

Aa

reused for border only
--main-border: 3px solid #1a252f
border: var(--main-border)

Aa

reused for text color
--main-bg: #2c3e50
color: var(--main-bg)

Aa

literal (no var, for comparison) (no custom property used)
background: #2c3e50 (literal)

2 — Fallback Values

Each box references a custom property that was never declared; the fallback (second argument to var()) is used instead.

Aa

color fallback
--undefined-bg: (not declared) → fallback #8e44ad
background: var(--undefined-bg, #8e44ad)

Aa

length fallback
--undefined-padding: (not declared) → fallback 16px
padding: var(--undefined-padding, 16px)

Aa

nested fallback chain
--undefined-a, --undefined-b: (not declared) → fallback #d35400
var(--a, var(--b, #d35400))

Aa

no fallback (uses initial)
--totally-undefined: (not declared) → initial value
background: var(--totally-undefined)

3 — Inheritance & Local Override

--accent is declared once on the outer container. The second box overrides it locally; the override does not leak to its siblings.

Aa

inherited (no override)
--accent: #2980b9 (inherited)
background: var(--accent) → inherited

Aa

local override
--accent: #c0392b (local override)
--accent: #c0392b (local)

Aa

sibling still inherits original
--accent: #2980b9 (inherited, unaffected)
background: var(--accent) → unaffected

Aa

--accent: unset (still inherits)
--accent: #2980b9 (via unset → inherit)
--accent: unset; background: var(--accent)

4 — Cyclic References Resolve Safely

Per spec, a custom property that references itself (directly or through a chain) becomes invalid instead of looping forever; a fallback or the property's initial value is used.

Aa

direct cycle, with fallback
--loop-a → --loop-b: cyclic → invalid → fallback #7f8c8d
--loop-a: var(--loop-b); --loop-b: var(--loop-a); background: var(--loop-a, #7f8c8d)

Aa

self-reference — always invalid
--self: self-referential → invalid (the fallback inside --self's OWN definition does not rescue it — matches Chrome/Firefox)
--self: var(--self, #e74c3c); background: var(--self) (no fallback here)

Aa

one-directional chain (not cyclic)
--chain-a → --chain-b: #27ae60 (resolved, not cyclic)
--a: var(--b); --b: #27ae60 → resolves

Aa

multi-hop chain
--x1 → --x2 → --x3: #f39c12 (resolved)
--x1 → --x2 → --x3: #f39c12

5 — Real-World Example: Themeable Card Component

The same card markup and CSS rules render two different themes purely by changing custom property values on the wrapping element — no duplicated rules.

Light Theme

This card uses the component's default custom property values.

Learn More

--card-bg: white · --card-fg: #222 · --card-accent: #2c3e50 · --card-muted: #666 (defaults)

Dark Theme

Same markup and rules — only the custom property values differ.

Learn More

--card-bg: #1a1a2e · --card-fg: #eee · --card-accent: #e94560 · --card-muted: #aaa (overridden inline)