UI GUIDELINE

Last updated Feb 2026

Number Input *Number Input*

You may find the Number Input in several presentations, either as an Input, or a Label with increment and decrement buttons next to it. This component requires that the user less effort than selecting the input field, tapping the digit “2” on the keypad, and hitting Enter or dismissing the keyboard. For example, to increase the number of guests from 1 to 2 in a form, the user only needs one tap on the plus button.

CATEGORY

Inputs

SYSTEMS ANALYZED

11 of 20

design systems include this component

01

Naming Convention

How the top 11 design systems name this component. Use the recommended name for maximum interoperability.

RECOMMENDED NAME

Number Input

36% of systems (4/11)

36%

ADOPTION

ALTERNATIVE 1

Input Number

18% of systems (2/11)

18%

ADOPTION

ALTERNATIVE 2

Number Field

18% of systems (2/11)

18%

ADOPTION

ALTERNATIVE 3

Numeric Input

9% of systems (1/11)

9%

ADOPTION

ALTERNATIVE 4

Spin Button

9% of systems (1/11)

9%

ADOPTION

ALTERNATIVE 5

Stepper Input

9% of systems (1/11)

9%

ADOPTION

Who uses "Number Input"

Zenith Design | Coral UI | Terra Components | Flux DS

Who uses "Input Number"

Amber DS | Horizon UI

Who uses "Number Field"

Atlas Design | Nova Components

Who uses "Numeric Input"

Pine UI

Who uses "Spin Button"

Cobalt DS

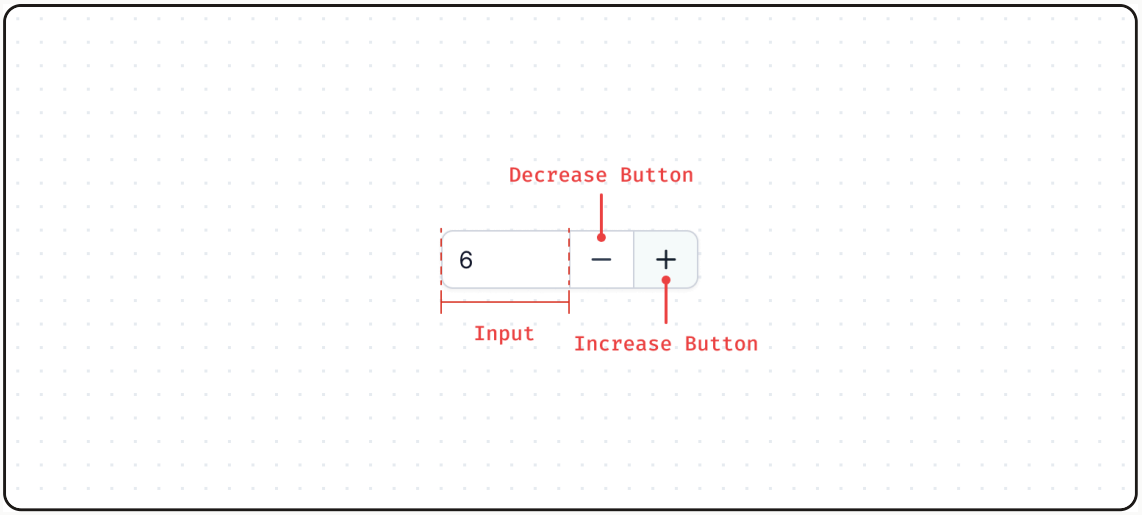
Who uses "Stepper Input"

Prism Design

02

Standard Anatomy

The standard structural parts that compose this component, derived from patterns observed across the 11 design systems analyzed. Use this as the foundation for your implementation.



03

Standard Props

Props found across the top 11 design systems, ordered by adoption frequency. Use this as a baseline for your API.

PROP	TYPE	DEFAULT	ADOPTION	DESCRIPTION
value	number	–	82%	The current value of the input.
min	number	–	73%	The minimum value of the input.
max	number	100	73%	The maximum value of the input.
size	"xs" | "sm" | "md" | "lg"	md	64%	Changes the size of the input, giving it more or less padding.
step	number	1	64%	Specify how much the value should increase / decrease upon clicking on up / down button.
precision	number	–	36%	How many decimal places the value should be rounded to.
disabled	boolean	false	91%	Prevents user interaction with the input.
readOnly	boolean	false	36%	Makes the input read-only, preventing value changes.
placeholder	string	–	45%	Hint text displayed when the input is empty.
onChange	function	–	55%	Called every time the input value changes.

04

Figma Build Checklist

Use this checklist when building this component in Figma. Check off each item as you design it.

What to design for Number Input

Design all size, state, and content variations needed for a production-ready Number Input component.

% = adoption across top design systems

SIZES

- [ ] **Small (sm)** 73%%
  Compact variant for dense forms and data tables
- [ ] **Medium (md)** 91%%
  Default size for standard forms — most commonly used
- [ ] **Large (lg)** 64%%
  Touch-friendly size for mobile-first or prominent inputs

STATES

- [ ] **Default** 100%%
  Resting state with placeholder or current value
- [ ] **Focused** 91%%
  Active keyboard focus — show focus ring and optionally highlight buttons
- [ ] **Hover** 82%%
  Subtle border or background change on mouse hover
- [ ] **Disabled** 91%%
  Non-interactive — reduced opacity, no pointer events
- [ ] **Read-only** 55%%
  Selectable but not editable — no increment/decrement buttons
- [ ] **Error / Invalid** 73%%
  Red border + error icon + message slot below input
- [ ] **At min boundary** 64%%
  Decrement button disabled/muted when value equals min
- [ ] **At max boundary** 64%%
  Increment button disabled/muted when value equals max

CONTENT VARIATIONS

- [ ] **With label** 82%%
  External label above or to the left of the input
- [ ] **With helper text** 64%%
  Hint text below the input explaining constraints (e.g. 'Min 1, Max 100')
- [ ] **With prefix/suffix** 45%%
  Units like '$', '%', 'kg' adjacent to the value
- [ ] **Without stepper buttons** 36%%
  Plain numeric input — no increment/decrement buttons visible

SUB-VARIANTS

- [ ] **Inline stepper** 73%%
  Buttons inside the input on left/right edges (most common pattern)
- [ ] **Side stepper** 36%%
  Buttons outside the input — separated visually

- [ ] **Compact / icon-only** 18%%
  Minimal footprint with small +/- icons, no visible input border

RESPONSIVE

- [ ] **Full-width on mobile** 55%%
  Input stretches to container width on small screens
- [ ] **Touch-target size** 45%%
  Stepper buttons meet 44×44px minimum tap target on touch devices

> **Don’t over-design**
> Start with Medium size, Default/Focused/Disabled/Error states, and Inline stepper layout. Add Read-only, boundary states, and size variants once the core is validated.

05

Keyboard Interactions

Based on the WAI-ARIA `spinbutton` pattern. Design focus and active states for each interaction.

KEY	ACTION	DESIGN NOTE
`ArrowUp`	Increases the value by step amount	*Show pressed/active state on increment button*
`ArrowDown`	Decreases the value by step amount	*Show pressed/active state on decrement button*
`Home`	Sets the value to minimum	*Value snaps to min — ensure it fits within the input field*
`End`	Sets the value to maximum	*Value snaps to max — ensure large numbers don't overflow*
`PageUp`	Increases the value by a larger step (10×)	*No extra visual state needed — same as ArrowUp interaction*
`PageDown`	Decreases the value by a larger step (10×)	*No extra visual state needed — same as ArrowDown interaction*

06

Accessibility

This component uses the `spinbutton` role — A numeric input that allows incrementing and decrementing values via keyboard or buttons.

🎨 YOUR DESIGN TASKS

Implement these in your Figma component to ensure accessibility compliance.

WHAT TO DESIGN	WHY	SPEC
Visible focus ring	Keyboard users must see where focus is at all times	`2px solid outline, offset 2px, using your system's focus color`
Disabled state opacity	aria-disabled tells assistive tech, but sighted users need visual cue	`40–50% opacity or muted fill + no pointer cursor`
Error state indicator	aria-invalid marks the error; design must make it visually obvious	`Red border + error icon + error message text below input`
Min/max boundary feedback	When value reaches min or max, the corresponding button should look disabled	`Muted color + no hover effect on the boundary-side button`

Screen reader behavior to keep in mind

Announces: "[label], spinbutton, [value], min [min], max [max]"
On value change: Announces new value automatically via aria-valuenow
On boundary: Should not announce anything extra — button disabled state is sufficient

📋 DEVELOPER HANDOFF — COPY TO FIGMA

Paste these notes into your Figma file so your developers know exactly what to implement.

- ✓ Use role="spinbutton" on the input element (not the wrapper)
- ✓ Set aria-valuenow, aria-valuemin, aria-valuemax on the input
- ✓ Add aria-label or aria-labelledby pointing to the visible label
- ✓ Set aria-disabled="true" (not just the HTML disabled attribute) on the input when disabled
- ✓ Increment/decrement buttons must have aria-hidden="true" — they're redundant for screen readers who use arrow keys
- ✓ Add aria-invalid="true" and aria-describedby pointing to error message when validation fails

11

Figma UI Kits

Figma UI kits that include a design for this component.