

Getting Started

- Introduction
- Installation
- components.json
- Theming
- Dark mode
- CLI
- Monorepo 

New
- Next.js 15 + React 19
- Typography
- Open in v0
- Figma
- Changelog

Installation

- Next.js
- Vite
- Remix
- Astro
- Laravel
- Gatsby
- Manual

Components

- Accordion
- Alert
- Alert Dialog
- Aspect Ratio
- Avatar
- Badge
- Breadcrumb
- Button
- Calendar
- Card
- Carousel
- Chart

Docs > Input OTP

# Input OTP

Accessible one-time password component with copy paste functionality.

Docs [🔗](#)

PreviewCode

Style: New York

Open in

—

## About

Input OTP is built on top of [input-otp](#) by [@guilherme\\_rodz](#).

## Installation

CLIManual

1 Run the following command:

```
pnpm npm yarn bun

pnpm dlx shadcn@latest add input-otp
```

2 Update tailwind.config.js

Add the following animations to your tailwind.config.js file:

```
tailwind.config.js

1 /** @type {import('tailwindcss').Config} */
2 module.exports = {
3   theme: {
4     extend: {
5       keyframes: {
6         "caret-blink": {
7           "0%,70%,100%": { opacity: "1" },
8           "20%,50%": { opacity: "0" },
9         },
10      },
11      animation: {
12        "caret-blink": "caret-blink 1.25s ease-out infinite",
13      },
14    },
15  },
16 }
```

## Usage

```
import {
  InputOTP,
  InputOTPGroup,
  InputOTPSeparator,
  InputOTPSlot,
} from "@components/ui/input-otp"
```

```
<InputOTP maxLength={6}>
  <InputOTPGroup>
    <InputOTPSlot index={0} />
    <InputOTPSlot index={1} />
    <InputOTPSlot index={2} />
  </InputOTPGroup>
  <InputOTPSeparator />
  <InputOTPGroup>
    <InputOTPSlot index={3} />
    <InputOTPSlot index={4} />
    <InputOTPSlot index={5} />
  </InputOTPGroup>
</InputOTP>
```

## Examples

### Pattern

Use the pattern prop to define a custom pattern for the OTP input.

PreviewCode

Style: New York

Open in

```
1 import { REGEXP_ONLY_DIGITS_AND_CHARS } from "input-otp"
2
3 ...
4
5 <InputOTP
6   maxLength={6}
7   pattern={REGEXP_ONLY_DIGITS_AND_CHARS}
8 >
9   <InputOTPGroup>
10     <InputOTPSlot index={0} />
11     {/* ... */}
12   </InputOTPGroup>
13 </InputOTP>
```

### Separator

You can use the <InputOTPSeparator /> component to add a separator between the input groups.

PreviewCode

Style: New York

Open in

—

—

```
1 import {
2   InputOTP,
3   InputOTPGroup,
4   InputOTPSeparator,
5   InputOTPSlot,
6 } from "@components/ui/input-otp"
7
8 ...
9
10 <InputOTP maxLength={4}>
11   <InputOTPGroup>
12     <InputOTPSlot index={0} />
13     <InputOTPSlot index={1} />
14   </InputOTPGroup>
15   <InputOTPSeparator />
16   <InputOTPGroup>
17     <InputOTPSlot index={2} />
18     <InputOTPSlot index={3} />
19   </InputOTPGroup>
20 </InputOTP>
```

### Controlled

You can use the value and onChange props to control the input value.

PreviewCode

Style: New York

Open in

Enter your one-time password.

## Form

PreviewCode

Style: New York

Open in

One-Time Password

Please enter the one-time password sent to your phone.

Submit

## Changelog

### 2024-03-19 Composition

We've made some updates and replaced the render props pattern with composition. Here's how to update your code if you prefer the composition pattern.

**Note:** You are not required to update your code if you are using the render prop. It is still supported.

1 Update to the latest version of input-otp .

```
pnpm npm yarn bun

pnpm add input-otp@latest
```

2 Update input-otp.tsx

```
input-otp.tsx

1 - import { OTPInput, SlotProps } from "input-otp"
2 + import { OTPInput, OTPInputContext } from "input-otp"
3
4 const InputOTPSlot = React.forwardRef<
5   React.ElementRef<"div">,
6   React.ComponentPropsWithoutRef<"div">
7   >(({ char, hasFakeCaret, isActive, className, ...props }, ref) => {
8     + React.ComponentPropsWithoutRef<"div"> & {
9       index: number }
10    + >(({ index, className, ...props }, ref) => {
11      + const inputOTPContext =
12        React.useContext(OTPInputContext)
13      + const { char, hasFakeCaret, isActive } =
14        inputOTPContext.slots[index]
15    })
16  })
```

3 Then replace the render prop in your code.

```
1 <InputOTP maxLength={6}>
2   <InputOTPGroup>
3     <InputOTPSlot index={0} />
4     <InputOTPSlot index={1} />
5     <InputOTPSlot index={2} />
6   </InputOTPGroup>
7   <InputOTPSeparator />
8   <InputOTPGroup>
9     <InputOTPSlot index={3} />
10    <InputOTPSlot index={4} />
11    <InputOTPSlot index={5} />
12  </InputOTPGroup>
13 </InputOTP>
```

### 2024-03-19 Disabled

To add a disabled state to the input, update <InputOTP /> as follows:

```
input-otp.tsx

1 const InputOTP = React.forwardRef<
2   React.ElementRef<typeof OTPInput>,
3   React.ComponentPropsWithoutRef<typeof OTPInput>
4 >(({ className, containerClassName, ...props }, ref)
5 => {
6   <OTPInput
7     ref={ref}
8     containerClassName={cn(
9       "flex items-center gap-2 has-[:disabled]:opacity-50",
10      containerClassName
11    )}
12     className={cn("disabled:cursor-not-allowed",
13      className)}
14     {...props}
15   />
16 })
17 InputOTP.displayName = "InputOTP"
```