

Checkbox Component

The Checkbox component provides a toggleable input control that allows users to select one or multiple options. It supports various states and styles while maintaining accessibility.

Import

```
import { Checkbox } from "@components/ui/checkbox"
```

API Reference

Props

- `checked?: boolean` - Controlled checked state
- `defaultChecked?: boolean` - Default checked state
- `onCheckedChange?: (checked: boolean) => void` - Change handler
- `disabled?: boolean` - Disabled state
- `required?: boolean` - Required state
- All HTML input props are supported

Styling

Base Styles

```
.checkbox {
  height: var(--checkbox-size);
  width: var(--checkbox-size);
  border-radius: var(--radius-sm);
  border: 1px solid var(--color-border);
  background-color: var(--color-surface);
  transition: all var(--duration-fast) var(--ease-out);
}

.checkbox[data-state="checked"] {
  background-color: var(--color-primary);
  border-color: var(--color-primary);
}
```

States

Basic States

```
// Default
<Checkbox />

// Checked
<Checkbox checked />

// Disabled
<Checkbox disabled />

// Required
<Checkbox required />

// Indeterminate
<Checkbox checked="indeterminate" />
```

Examples

Basic Checkbox

```
<div className="flex items-center space-x-2">
  <Checkbox id="terms" />
  <label
    htmlFor="terms"
    className="text-sm font-medium leading-none"
  >
    Accept terms and conditions
  </label>
</div>
```

Checkbox Group

```
function CheckboxGroup() {
  const [selected, setSelected] = useState<string[]>([])

  return (
    <div className="space-y-2">
      <Checkbox
        checked={selected.includes("option1")}
        onCheckedChange={(checked) => {
          if (checked) {
            setSelected([...selected, "option1"])
          } else {
            setSelected(selected.filter(x => x !== "option1"))
          }
        }}
      />
      Option 1
    </div>
  )
}
```

```

    </Checkbox>
    { /* More options */ }
  </div>
)
}

```

With Helper Text

```

<div className="space-y-2">
  <div className="flex items-center space-x-2">
    <Checkbox id="emails" />
    <label htmlFor="emails">
      Receive marketing emails
    </label>
  </div>
  <p className="text-sm text-foreground-muted pl-6">
    Get notified about new products and offers.
  </p>
</div>

```

Animations

Check Animation

```

.checkbox-indicator {
  transform-origin: center;
  animation: checkbox-pop var(--duration-fast) var(--ease-out);
}

@keyframes checkbox-pop {
  from {
    transform: scale(0.8);
    opacity: 0;
  }
  to {
    transform: scale(1);
    opacity: 1;
  }
}

```

Accessibility

Keyboard Navigation

- **Space**: Toggle checkbox
- **Tab**: Move focus between checkboxes

- **Shift + Tab**: Move focus in reverse

ARIA Support

```
<div role="group" aria-labelledby="group-label">
  <span id="group-label">Select options:</span>
  <Checkbox
    aria-label="Option 1"
    aria-describedby="option1-description"
  />
  <span id="option1-description">
    Detailed description of option 1
  </span>
</div>
```

Integration

With Forms

```
import { useForm } from "react-hook-form"

function CheckboxForm() {
  const form = useForm({
    defaultValues: {
      marketing: false,
    },
  })

  return (
    <form onSubmit={form.handleSubmit(onSubmit)}>
      <Controller
        control={form.control}
        name="marketing"
        render={({ field }) => (
          <Checkbox
            checked={field.value}
            onChange={field.onChange}
          />
        )}
      />
    </form>
  )
}
```

With Custom Styles

```
<Checkbox
  className="data-[state=checked]:bg-green-500
    data-[state=checked]:border-green-500"
/>
```

Design Tokens

```
:root {
  /* Checkbox specific tokens */
  --checkbox-size: 1rem;
  --checkbox-icon-size: 0.875rem;

  --checkbox-border-color: var(--color-border);
  --checkbox-background: var(--color-surface);
  --checkbox-checked-background: var(--color-primary);
  --checkbox-checked-color: var(--color-background);

  --checkbox-transition-duration: var(--duration-fast);
  --checkbox-transition-timing: var(--ease-out);
}
```

Troubleshooting

Common Issues

1. Checkbox not aligned with label

```
// Solution: Use proper flexbox alignment
<div className="flex items-center">
  <Checkbox />
  <label className="ml-2 leading-none">Label</label>
</div>
```

2. Custom styles not applying

```
// Solution: Use data attributes for state-based styling
<Checkbox className="
  data-[state=checked]:bg-brand
  data-[state=checked]:border-brand
" />
```

3. Form integration issues

```
// Solution: Use controlled component pattern
const [checked, setChecked] = useState(false)

<Checkbox
  checked={checked}
  onCheckedChange={setChecked}
/>
```

Best Practices

1. Label Association

- Always provide labels for checkboxes
- Use semantic HTML for labels
- Ensure proper label alignment

2. Interaction Design

- Provide clear hover and focus states
- Include proper spacing for touch targets
- Use animation for state changes

3. Form Usage

- Group related checkboxes
- Provide clear group labels
- Handle form submission properly

4. Accessibility

- Use proper ARIA attributes
- Support keyboard navigation
- Provide error states and messages

Variants

Size Variants

```
// Small
<Checkbox className="h-4 w-4" />

// Default
<Checkbox />

// Large
<Checkbox className="h-6 w-6" />
```

Custom Indicators

```
<Checkbox>
  <CheckboxIndicator>
    <CustomIcon className="h-4 w-4" />
  </CheckboxIndicator>
</Checkbox>
```

Error State

```
<div>
  <div className="flex items-center">
    <Checkbox
      className="border-red-500 data-[state=checked]:bg-red-500"
    />
    <label className="text-red-500">Required field</label>
  </div>
  <p className="text-red-500 text-sm mt-1">
    This field is required
  </p>
</div>
```

Would you like me to provide any additional details or examples for specific aspects of the Checkbox component?