

# Select Component

---

The Select component provides a dropdown menu for selecting one value from a list of options. Built on Radix UI's Select primitive, it supports various states, sizes, and grouping while maintaining accessibility.

## Import

```
import {
  Select,
  SelectGroup,
  SelectValue,
  SelectTrigger,
  SelectContent,
  SelectLabel,
  SelectItem,
  SelectSeparator,
} from "@components/ui/select"
```

## API Reference

### Props

#### Select

- `value?: string` - Controlled value
- `defaultValue?: string` - Default value
- `onValueChange?: (value: string) => void` - Value change handler
- `disabled?: boolean` - Disabled state

#### SelectTrigger

- `className?: string` - Additional CSS classes
- `children: React.ReactNode` - Trigger content

#### SelectContent

- `position?: "popper" | "item-aligned"` - Content position strategy
- `className?: string` - Additional CSS classes

#### SelectItem

- `value: string` - Item value
- `disabled?: boolean` - Disabled state
- `className?: string` - Additional CSS classes

# Styling

## Base Styles

```
.select-trigger {  
  height: var(--select-trigger-height);  
  padding: var(--space-2) var(--space-3);  
  background-color: var(--color-surface);  
  border: 1px solid var(--color-border);  
  border-radius: var(--radius-md);  
  color: var(--color-text-primary);  
  transition: var(--select-transition);  
}  
  
.select-content {  
  background-color: var(--color-surface);  
  border: 1px solid var(--color-border);  
  border-radius: var(--radius-md);  
  box-shadow: var(--shadow-md);  
}
```

## States

### Basic States

```
// Default  
<Select>  
  <SelectTrigger>  
    <SelectValue placeholder="Select option" />  
  </SelectTrigger>  
  <SelectContent>  
    <SelectItem value="option1">Option 1</SelectItem>  
  </SelectContent>  
</Select>  
  
// Disabled  
<Select disabled>  
  <SelectTrigger>  
    <SelectValue placeholder="Disabled" />  
  </SelectTrigger>  
</Select>  
  
// Error  
<Select>  
  <SelectTrigger className="select-error">  
    <SelectValue placeholder="Error state" />  
  </SelectTrigger>  
</Select>
```

## Examples

### Basic Select

```
<Select>
  <SelectTrigger>
    <SelectValue placeholder="Theme" />
  </SelectTrigger>
  <SelectContent>
    <SelectItem value="light">Light</SelectItem>
    <SelectItem value="dark">Dark</SelectItem>
    <SelectItem value="system">System</SelectItem>
  </SelectContent>
</Select>
```

### With Groups

```
<Select>
  <SelectTrigger>
    <SelectValue placeholder="Select a fruit" />
  </SelectTrigger>
  <SelectContent>
    <SelectGroup>
      <SelectLabel>Fruits</SelectLabel>
      <SelectItem value="apple">Apple</SelectItem>
      <SelectItem value="banana">Banana</SelectItem>
    </SelectGroup>
    <SelectSeparator />
    <SelectGroup>
      <SelectLabel>Vegetables</SelectLabel>
      <SelectItem value="carrot">Carrot</SelectItem>
      <SelectItem value="potato">Potato</SelectItem>
    </SelectGroup>
  </SelectContent>
</Select>
```

## Animations

### Animation Tokens

```
:root {
  --select-transition: all var(--duration-fast) var(--ease-out);
}

.select-content-enter {
```

```

    animation: select-content-in var(--duration-normal) var(--ease-out);
}

@keyframes select-content-in {
  from {
    opacity: 0;
    transform: translateY(-4px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}

```

## Accessibility

### Keyboard Navigation

- **Space/Enter**: Open/close select menu
- **Up/Down**: Navigate through options
- **Home/End**: Jump to first/last option
- **Escape**: Close select menu
- **Tab**: Move focus to next element

### ARIA Support

```

<Select aria-label="Theme selection">
  <SelectTrigger aria-label="Select theme">
    <SelectValue />
  </SelectTrigger>
  <SelectContent>
    <SelectGroup aria-label="Theme options">
      <SelectItem value="light">Light theme</SelectItem>
    </SelectGroup>
  </SelectContent>
</Select>

```

## Integration

### With Forms

```

import { useForm, Controller } from "react-hook-form"

function SelectForm() {
  const form = useForm({
    defaultValues: {
      theme: "",

```

```

    },
  })

  return (
    <form onSubmit={form.handleSubmit(onSubmit)}>
      <Controller
        control={form.control}
        name="theme"
        render={({ field }) => (
          <Select
            onChange={field.onChange}
            defaultValue={field.value}
          >
            <SelectTrigger>
              <SelectValue placeholder="Select theme" />
            </SelectTrigger>
            <SelectContent>
              <SelectItem value="light">Light</SelectItem>
              <SelectItem value="dark">Dark</SelectItem>
            </SelectContent>
          </Select>
        )}
      />
    </form>
  )
}

```

## Design Tokens

```

:root {
  /* Select specific tokens */
  --select-trigger-height: 2.5rem;
  --select-content-max-height: 24rem;
  --select-item-height: 2rem;

  /* Content dimensions */
  --select-min-width: 8rem;
  --select-padding: var(--space-1);

  /* Indicators */
  --select-indicator-size: 1rem;
  --select-check-size: 0.875rem;

  /* Transitions */
  --select-transition: all var(--duration-fast) var(--ease-out);

  /* Z-index */
  --select-z-index: var(--z-dropdown);
}

```

# Troubleshooting

## Common Issues

### 1. Select not opening

```
// Solution: Ensure trigger is properly connected
<Select>
  <SelectTrigger /> {/* Must be direct child of Select */}
  <SelectContent />
</Select>
```

### 2. Value not updating

```
// Solution: Use controlled component pattern
const [value, setValue] = useState("")

<Select value={value} onChange={setValue}>
  <SelectTrigger>
    <SelectValue />
  </SelectTrigger>
</Select>
```

### 3. Positioning issues

```
// Solution: Use position prop
<SelectContent position="popper" sideOffset={4}>
  {/* Content */}
</SelectContent>
```

## Best Practices

### 1. Label Usage

- Always provide clear labels
- Use descriptive placeholder text
- Include helper text when needed

### 2. Grouping

- Group related options together
- Use separators for clarity
- Provide group labels

### 3. Accessibility

- Include proper ARIA labels
- Support keyboard navigation
- Maintain focus management

#### 4. Performance

- Virtualize large option lists
- Implement proper loading states
- Handle async data loading

## Variants

### Size Variants

```
// Small
<Select>
  <SelectTrigger className="select-sm">
    <SelectValue />
  </SelectTrigger>
</Select>

// Large
<Select>
  <SelectTrigger className="select-lg">
    <SelectValue />
  </SelectTrigger>
</Select>
```

### With Helper Text

```
<div className="space-y-2">
  <Select>
    <SelectTrigger>
      <SelectValue placeholder="Select role" />
    </SelectTrigger>
    <SelectContent>
      {/* Options */}
    </SelectContent>
  </Select>
  <p className="text-sm text-foreground-muted">
    Select your role in the organization
  </p>
</div>
```

### With Loading State

```
<Select disabled>
  <SelectTrigger className="select-loading">
    <SelectValue placeholder="Loading..." />
  </SelectTrigger>
</Select>
```