

Tooltip

Dicas exibem texto informativo quando os usuários passam o mouse, focalizam ou tocam em um elemento.

Quando ativada, [dicas](#) exibem um rótulo de texto identificando o elemento, como uma descrição de sua função.

```
{{"component": "modules/components/ComponentLinkHeader.js"}}
```

Dicas simples

```
{{"demo": "BasicTooltip.js"}}
```

Dicas posicionadas

O `Tooltip` tem 12 **posicionamentos** para ser escolhido. They don't have directional arrows; instead, they rely on motion emanating from the source to convey direction.

```
{{"demo": "PositionedTooltips.js"}}
```

Dicas customizadas

Aqui estão alguns exemplos de customização do componente. Você pode aprender mais sobre isso na [página de documentação de sobrescritas](#).

```
{{"demo": "CustomizedTooltips.js"}}
```

Dicas com seta

Você pode usar a propriedade `arrow` para dar à sua dica uma seta indicando a qual elemento se refere.

```
{{"demo": "ArrowTooltips.js"}}
```

Elemento filho customizado

The tooltip needs to apply DOM event listeners to its child element. If the child is a custom React element, you need to make sure that it spreads its props to the underlying DOM element.

```
const MyComponent = React.forwardRef(function MyComponent(props, ref) {
  // Distribua as propriedades para o elemento DOM subjacente.
  return <div {...props} ref={ref}>Bin</div>
});

// ...

<Tooltip title="Excluir">
  <MyComponent>
</Tooltip>
```

Você pode encontrar um conceito similar no guia [encapaulando componentes](#).

Gatilhos

Você pode definir os tipos de eventos que fazem com que uma dica seja exibida.

The touch action requires a long press due to the `enterTouchDelay` prop being set to `700 ms` by default.

```
{{"demo": "TriggersTooltips.js"}}
```

Dicas controladas

You can use the `open`, `onOpen` and `onClose` props to control the behavior of the tooltip.

```
{{"demo": "ControlledTooltips.js"}}
```

Largura variável

A dica (`Tooltip`) quebra o texto longo por padrão para torná-lo legível.

```
{{"demo": "VariableWidth.js"}}
```


Interativo

Tooltips are interactive by default (to pass [WCAG 2.1 success criterion 1.4.13](#)). Ela não será fechada quando o usuário passar por cima da dica antes que `leaveDelay` expire. You can disable this behavior (thus failing the success criterion which is required to reach level AA) by passing `disableInteractive`.

```
{{"demo": "NonInteractiveTooltips.js"}}
```

Elementos desabilitados

Por padrão os elementos desativados como `<button>` não disparam interações do usuário, então uma `Tooltip` não será ativada em eventos normais, como passar o mouse. Para acomodar elementos desabilitados, adicione um elemento encapsulador simples, como um `span`.

 Para trabalhar com o Safari, você precisa de pelo menos um `display block` ou `flex item` abaixo do elemento que encapsula a dica.

```
{{"demo": "DisabledTooltips.js"}}
```

Se você não estiver manipulando com um componente Material-UI que herde de `ButtonBase`, por exemplo, um elemento `<button>` nativo, você também deve adicionar a propriedade CSS `pointer-events: none;` ao seu elemento quando desabilitado:

```
<Tooltip title="You don't have permission to do this">
  <span>
    <button disabled={disabled} style={disabled ? { pointerEvents: 'none' } : {}}>
      A disabled button
    </button>
  </span>
</Tooltip>
```

Transições

Use a different transition.

```
{{"demo": "TransitionsTooltips.js"}}
```

Seguir o cursor

You can enable the tooltip to follow the cursor by setting `followCursor={true}` .

```
{{"demo": "FollowCursorTooltips.js"}}
```

Elemento virtual

In the event you need to implement a custom placement, you can use the `anchorEl` prop: The value of the `anchorEl` prop can be a reference to a fake DOM element. You need to create an object shaped like the [VirtualElement](#) .

```
{{"demo": "AnchorElTooltips.js"}}
```

Exibindo e ocultando

A dica normalmente é mostrada imediatamente quando o mouse do usuário passa sobre o elemento e se oculta imediatamente quando o mouse do usuário sai. A delay in showing or hiding the tooltip can be added through the `enterDelay` and `leaveDelay` props, as shown in the Controlled Tooltips demo above.

No celular, a dica é exibida quando o usuário pressiona longamente o elemento e oculta após um atraso de 1500 ms. You can disable this feature with the `disableTouchListener` prop.

```
{{"demo": "DelayTooltips.js"}}
```

Acessibilidade

(WAI-ARIA: <https://www.w3.org/TR/wai-aria-practices/#tooltip>)

By default, the tooltip only labels its child element. This is notably different from `title` which can either label **or** describe its child depending on whether the child already has a label. For example, in:

```
<button title="alguma informação a mais">Um botão</button>
```

the `title` acts as an accessible description. If you want the tooltip to act as an accessible description you can pass `describeChild` . Note that you shouldn't use `describeChild` if the tooltip provides the only visual label. Otherwise, the child would have no accessible name and the tooltip would violate [success criterion 2.5.3 in WCAG 2.1](#).

```
{{"demo": "AccessibilityTooltips.js"}}
```