

Componentes React para progresso circular, linear

Progresso

Indicadores de progresso comumente conhecidos como spinners, expressam um tempo de espera não especificado ou exibem a duração de um processo.

Indicadores de progresso informam aos usuários sobre o estado de processos em progresso, como o carregamento de um aplicativo, envio de um formulário, ou atualizações.

- O indicador circular **determinado** preenche a faixa circular invisível com cor, a medida que o indicador se move de 0 a 360 graus.
- Indicador **indeterminado** demonstra um tempo de espera não especificado.

Indicador **determinado** mostra quanto tempo uma operação vai demorar.

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

Circular

Circular indeterminado

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

Circular color

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

Circular determinado

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

Integração interativa

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

Circular com rótulo

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

Linear

Linear indeterminado

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

Linear color

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

Linear determinado

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

Linear bufferizado

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

Linear com rótulo

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

Intervalo não-padrão

Os componentes de progresso aceitam um valor no intervalo de 0 a 100. Isso simplifica as coisas para os usuários de leitores de tela, onde estes são os valores padrão mínimos / máximos. Às vezes, no entanto, você pode estar trabalhando com uma fonte de dados onde os valores ficam de fora deste intervalo. Veja como você pode transformar facilmente um valor em qualquer intervalo em uma escala de 0 a 100:

```
// MIN = Valor mínimo esperado
// MAX = Valor máximo esperado
// Função para normalizar os valores (MIN / MAX pode ser integrado)
const normalise = (value) => ((value - MIN) * 100) / (MAX - MIN);

// Exemplo de componente que utiliza a função `normalise` no ponto de renderização.
function Progress(props) {
  return (
    <React.Fragment>
      <CircularProgress variant="determinate" value={normalise(props.value)} />
      <LinearProgress variant="determinate" value={normalise(props.value)} />
    </React.Fragment>
  );
}
```

Progresso customizado

Aqui estão alguns exemplos de customização do componente. Você pode aprender mais sobre isso na página de documentação de sobrescritas.

```
{{“demo”: “CustomizedProgressBars.js”, “defaultCodeOpen”: false}}
```

Aparência atrasada

Existem 3 limites importantes para saber sobre tempo de resposta. O efeito cascata do componente **ButtonBase** garante que o usuário experimente um sistema de feedback em tempo real. Normalmente, nenhum feedback é necessário durante atrasos de mais de 0,1 segundos e menos de 1,0 segundos. Mas após 1,0 segundo, você pode exibir um carregador para garantir que o fluxo de pensamento do usuário não seja interrompido.

```
{{“demo”: “DelayingAppearance.js”}}
```

Limitações

Alto carregamento da CPU

Quando o processamento é particularmente lento, você pode perder a animação do traço ou ver raios aleatórios com **CircularProgress**. Para não bloquear o processo principal de renderização, você deve processar suas operações com um web worker ou por batch.

carga pesada

Quando não for possível, você pode utilizar a propriedade **disableShrink** para atenuar o problema. Veja este problema.

```
{{“demo”: “CircularUnderLoad.js”}}
```

Atualizações de alta frequência

O **LinearProgress** usa uma transição na propriedade de transformação do CSS para fornecer uma atualização suave entre valores diferentes. A duração de transição padrão é de 200ms. Caso um componente pai atualize a propriedade **value** muito rapidamente, você irá perceber ao menos um atraso de 200ms entre a renderização e a barra de progresso totalmente atualizada.

Se você precisar executar 30 renderizações por segundo ou mais, recomendamos desabilitar a transição:

```
.MuiLinearProgress-bar {  
  transition: none;  
}
```

IE 11

The circular progress component animation on IE 11 is degraded. A animação do traço não está funcionando (equivalente a `disableShrink`) e a animação circular oscila. Você pode resolver o último com:

```
. MuiCircularProgress-indeterminate {  
  animation: circular-rotate 1.4s linear infinite;  
}  
  
@keyframes circular-rotate {  
  0% {  
    transform: rotate(0deg);  
    /* Corrige oscilações do IE11 */  
    transform-origin: 50% 50%;  
  }  
  100% {  
    transform: rotate(360deg);  
  }  
}
```