Progress 进度条组件

Progress indicators（进度指示器）也称为微调器 (spinners)，它表示了一个不明确的等待时间，或者显示处理过程的时间长短。

进度指示器能够将当前处理过程的状态通知用户，例如加载一个应用，提交一个表单或保存一些更新。

- **定量**指示器显示一个操作消耗多长时间。
- **不定量**指示器可视化了一个不确定的操作等待时间。

甚至在加载 JavaScript 之前，组件的动画就尽可能地依赖 CSS 来工作。

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

环状进度条

不定量的环形进度条

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

Circular color

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

定量的环形进度条

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

交互集成

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

带标签的环状进度条

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

线性进度条

不定量的线性进度条

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

Linear color

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

定量的线性进度条

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

线性缓冲条

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

带标签的线性进度条

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

非标准区间

进度条组件采用一个在 0 — 100 区间内的值。 作为默认的最小/最大值，屏幕阅读用户能够更便利地阅读。 但是有时，您可能会使用超出这个范围的数据源的一些值。 通过这个例子，您可以轻松地将一个任意区间的值转换为 0 — 100 区间的值：

```
// MIN = 最小期待值
// MAX = 最大期待值
// 使数值正常化的功能（MIN/MAX 可以整合）。 const normalise = (value) => ((value - MIN) * 100) / (MAX - MIN);

// 此示例组件在渲染时优化了“ normalise”函数。
function Progress(props) {
  return (
    <React.Fragment>
      <CircularProgress variant="determinate" value={normalise(props.value)} />
      <LinearProgress variant="determinate" value={normalise(props.value)} />
    </React.Fragment>
  );
}
```

定制的进度条

你可以参考以下一些例子来自定义组件。 您可以在 重写文档页面 中了解更多有关此内容的信息。

{{"demo": "CustomizedProgressBars.js", "defaultCodeOpen": false}}

延时的出现

关于响应时间，有 3 个重要限制。 `ButtonBase` 组件的波纹效果确保用户感受到系统是实时反馈的。 通常情况下，在多余 0.1 秒且小于 1.0 秒期间不需要特殊的反馈。 在 1.0 秒后，你可以显示一个加载器来保持用户的思考流程不被打断。

{{"demo": "DelayingAppearance.js"}}

设计局限

高负载

在加载特别慢时，你可能丢失 stroke dash 的动画或看到 `CircularProgress` 会有一些半径随机的情况。 为了不阻碍主渲染进程，您应该在 web worker 中或批处理中运行密集操作的处理器。

 高负载

当这个方法不可行的时候，您可以利用 `disableShrink` 属性来缓解这个问题。 请查看 这个问题。

{{"demo": "CircularUnderLoad.js"}}

高频更新

`LinearProgress` 在 CSS transform 属性上使用过渡参数来提供不同值之间的平滑更新。 默认过渡的持续时间为 200ms。 在父组件更新 `value` 属性过快的情况下，重新渲染和进度条完全更新之间至少会有 200ms 的延迟。

如果你需要每秒执行 30 次或更多次数的渲染，我们建议禁用过渡：

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

IE 11

IE 11 上的循环进度组件动画会退化。 Stroke dash 动画将不起作用（相当于 `disableShrink`），并且 circular 动画将会抖动。 你可以通过以下方式来解决后者：

```
.MuiCircularProgress-indeterminate {
  animation: circular-rotate 1.4s linear infinite;
}

@keyframes circular-rotate {
  0% {
    transform: rotate(0deg);
    /* 修复 IE11 下的抖动 */
    transform-origin: 50% 50%;
  }
  100% {
    transform: rotate(360deg);
  }
}
```