

```
app-tooltip.tsx
export class AppTooltip implements ComponentInterface {
  @State() tooltipVisible = false;
  onToggleTooltip() {
    this.tooltipVisible = !this.tooltipVisible;
  }
  render() {
    return [
      <slot></slot>,
      <span id="tooltip-icon" onClick={this.onToggleTooltip.bind(this)}></span>,
      <div id="tooltip-text">
        Hi! I am a tooltip, nice meeting you
      </div>
    ]
  }
}
```

```
index.html
<body>

  <p>Let's test this new <app-tooltip>Tooltip!</app-tooltip></p>

</body>
```

```
app-tooltip.css
#tooltip-icon {
  background-color: #fa1616;
  color: white;
  padding: 0.15rem 0.45rem;
  border-radius: 50%;
  margin-left: 0.5rem;
}
```

Show the tooltip text only when click on the ? icon

```
app-tooltip.tsx
export class AppTooltip implements ComponentInterface {
  @State() tooltipVisible = false;
  onToggleTooltip() {
    this.tooltipVisible = !this.tooltipVisible;
  }
  render() {
    let tooltip = null;
    if (this.tooltipVisible) {
      tooltip = <div id="tooltip-text">hi! I am a tooltip, nice meeting you</div>
    }

    return [
      <slot></slot>,
      <span id="tooltip-icon" onClick={this.onToggleTooltip.bind(this)}>?</span>,
      tooltip
    ]
  }
}
```

if it is visible set tooltip = text

display it here

```
app-tooltip.css
:host {
  display: inline;
  position: relative;
}

#tooltip-text {
  position: absolute;
  top: 1.5rem;
  left: 1rem;
  width: 10rem;
  background-color: #0fabbc;
  color: white;
  padding: 0.5rem;
  border-radius: 3px;
  box-shadow: 0 2px 6px rgba(0,0,0,0.26);
}
```

set position absolute to the information display by <slot>

Pass text from the parent using @Prop()

app-tooltip.tsx

```
export class AppTooltip implements ComponentInterface {
  @State() tooltipVisible = false;
  @Prop() text: string;
  onToggleTooltip() {
    this.tooltipVisible = !this.tooltipVisible;
  }
  render() {
    let tooltip = null;
    if (this.tooltipVisible) {
      tooltip = <div id="tooltip-text">{this.text}</div>
    }
    return [
      <slot></slot>,
      <span id="tooltip-icon" onClick={this.onToggleTooltip.bind(this)}>?</span>,
      tooltip
    ]
  }
}
```



A red arrow originates from the `text: string` prop in the `@Prop()` decorator and points down to the `{this.text}` interpolation within the `<div id="tooltip-text">` JSX element in the `render()` method.

use @Prop to pass text from the parent

index.html

```
<body>
  <p>Let's test this new
    <app-tooltip text="hi! I am a tooltip, very happy to see you">
      Tooltip!
    </app-tooltip>
  </p>
</body>
```



A red box highlights the `text="hi! I am a tooltip, very happy to see you"` attribute in the `<app-tooltip>` tag.