What are Control Props?

Here's a quick overview of control props!

WE'LL COVER THE FOLLOWING

Using Multiple `Expandable` Components

The compound component we built in the last section works great. However, it's quite limited in its extensibility.

Let me show you what I mean.

Assume another developer, who we'll refer to as "user", decided to use the component we've built, but in a much different way.

This user has found some very good information they think is worth sharing. Within their app, they store this information in an array:

```
// user's app
const information = [
  {
    header: 'Why everyone should live forever',
    note: 'This is highly sensitive information ... !!!!'
  },
  {
    header: 'The internet disappears',
    note:
      'I just uncovered the biggest threat...'
  },
  {
    header: 'The truth about Elon musk and Mars!',
    note: 'Nobody tells you this...'
  }
]
```

In their `App`, they loop over the list of information and render our `Expandable` component with its children as seen below:

```
.Expandable-panel {
    margin: 0;
    padding: 1em 1.5em;
    border: 1px solid hsl(216, 94%, 94%);;
    min-height: 150px;
  }
```

Nothing out of the ordinary here.

When the user clicks any of the headers, the contents expand as designed. Try clicking any one of the headers. One of the elements expands.

Expanding Multiple Headers at Once

But what happens when another header is clicked?

Well, that header expands as well. This is exactly how the internals of our `Expandable` component works.

As you can see from the output of the code above, both clicked headers expand their content. Unfortunately, this is not the behavior the user seeks.

What the user really wants is behavior such that only one container expands at a time. If the user clicks another header, it should open but the rest should close.

Quick Quiz!

1 What is the issue with the expandable component at the moment?

COMPLETED 0%

1 of 3

So, how do we extend the `Expandable` component to handle this use case?

Well, this is where the control props pattern comes into play. Note that I’m using the compound component we built as an example. In reality, you can implement this functionality in any component you wish. In the next lesson, we’ll learn how the control props pattern works.