Now that we can check off tasks and add tasks and read tasks, we're on to the very last thing we have

to implement which is how can we allow users to be able to delete tasks from the app? And I want to be

able to do that when a user simply touches and holds a particular item.

And as long as they've held it long enough

so a long touch,

then I want that item to delete off the list. So we know that our task is being provided right at the

top of our widget tree

and we know that that interaction is also being detected inside

our task tile, so low down in the tree.

How can we use what we know and what we've seen in the last lesson to be able to delete that task out

of our task list?

So in this case that behavior is gonna be detected right at the bottom of the tree in the task tile

and then it has to be able to take that information and give it over to the task list so that it's able

to update the task data and remove that task that the user selected. And then that information needs

to filter through the tree for all the interested parties, namely the task screen and the task list.

I recommend giving this one a go and try to implement that functionality and deleting that very item where

the long top is detected.

And if you get stuck along the way, just use Stack Overflow or search through Google and overcome those

hurdles that come your way.

And by the end, I promise you'll be a stronger developer for it.

And once you're done you can also come back and check with me where I'm gonna go through a version of

the solution. Pause the video now and give that a go.

The first thing we have to do is figure out well how we can actually detect that long press.

Now luckily for us, the list tile actually has a property called onLongPress which as it suggests, will

trigger a callback when it detects a long press,

so a couple of seconds. Now we can use exactly the same thing that we did with the checkbox callback.

We can create a new final function called a longPressCallback and we're also going to initialize it

inside the constructor

so right here. And we're going to pass it over to the onLongPress.

So when the list tile detects a long press, it's gonna trigger the callback and the callback is going

to come from the parent of the task tile which is of course our task list where our consumer exists

and where we do all of the updating to our provided task data.

So in addition to our checkbox callback, I'm now also going to add a longPressCallback. And in this

case what I'm going to trigger is simply a call to a delete method in our task data.

So I'm going to create another method called deleteTask and I'm going to pass over the task that needs

to be deleted. And then I'm going to tap into my list of tasks and I'm going to call the remove method

and I'm going to remove a particular object which is going to be the task that the user wanted to delete

which is passed in right here.

And of course I also need to notify all my listeners so that they can update as needed.

Now if I hit save over here and I go to my task list inside my long press callback,

so when somebody long presses on my task tile, what I want to happen is I want it to tap into my task

data,

call that method deleteTask and pass in that task which is the current task in the task tile.

So that's all I need to do.

And if I hit save and head over to my Todoey app and simply press and hold then, you'll see that my

items will get deleted. And it doesn't matter if it's completed or not completed doesn't matter if it's

higher up in the list, it all just works.

And it also updates right here where I'm listening for the number of tasks in my task list.

So that's it.

That's all we need to do to build our Todoey app which allows us to add tasks, checkoff tasks and

delete tasks.

So brilliant. We've managed to use provider to manage the state of our tasks and we've managed to learn

about a number of new things such as the list view builder or lifting state up and understand exactly

what state management is

anyways. Have a play around with the code and try and see if you can make some small tweaks or try to

extend the app so that you get more practice using state management with the provider package.

I hope you've had fun building this app with me and be sure to complete the questionnaire where you

can vote for the next topic you want to see in the course. And if there's enough students who are interested

in it, then I will build it.

So that's all from me for now and I'll see you on the next course that

you take.

