Now in the last lesson we managed to lift the state of our isChecked property up into a widget that's　higher up in the widget tree. And in order to pass an action that happened lower down in the tree we　used a callback to achieve it.

But how do callbacks actually work?

Well let's take a look at the code that we wrote in the last lesson in a little bit more detail.

Now this is a simplified version of our code so that it can fit onto the same screen because I really　want to show it to you side by side and I want to be able to animates some of the values of these properties　to show you how they flow through our code so we can better understand what's actually happening behind　the scenes.

Firstly notice up here I've got my task tile state class and this is the one that is stateful, so　it can change its property's values. And the property that we're most concerned about is this isChecked　property which is the one that we lifted up from the task checkbox into the parent widget which is the　task tile state.

And we did that to be able to use the value of that property to rebuild the text widget as well as our　checkbox widget.

So it's actually used right here.

So the value of this property is used when I create my task checkbox. So every time I rebuild my task　checkbox it looks to see the value of the isChecked property and it passes over into the constructor　and that goes then into the isChecked property inside my task checkbox.

So now that value can now be used inside the task checkbox and the purpose for it is to determine the　state of my checkbox widget whether if that box should be empty or whether if it should show a little　tick mark.

So when the value property is false it's going to be an empty square, when it's true then it's gonna　show the check mark. So that's easy enough to understand.

But the difficult part is how did we manage to get the user interaction which remember lives in the　checkbox widget which is in a downstream child widget of the task tile state because we first have the　task tile state and then we have the task checkbox.

So how did we manage to pass this information up the tree to the parent widget?

Well it's actually not quite passing data up. All that we're doing is we're saying well here is a callback　method that we've created called toggle check state and we're going to pass over this method and its　implementation into the task checkbox when it gets created.

So in exactly the same way as we did for the isChecked property, we're passing this function into this　task checkbox stateless widget so that we can use this function inside the onChanged property.

So that means when this checkbox changes then is going to trigger this callback and pass in the latest　version of the current state of the checkbox. So that callback is sitting there waiting to jump into　action and the event that it's waiting for is the moment when a user taps on that checkbox because that's　the moment when it's going to trigger its onChanged property and it's gonna call back this method and　pass in the current state of the checkbox.

So that method gets triggered so we can update the isChecked property to the latest value that got　passed over and we now have true. And our checkbox as well as all text widget are both able to update　the state based on this property. So if you imagine let's say that we had an App Brewery travel agent　and there was A which is us and then B which is you.

And in order for you to book a holiday with us,　well we actually have to fill in this really long form. And in order for this form to be filled in　we need a couple of pieces of information from you.

So we give you a quick call and we send the form over to you.

Now you take a look at this form and you see that there's a couple of pieces information in there that　you don't actually know　for example your passport number.

So you have to go and search for your passport around your house.

So we have to wait for you to go home from work and then look through your drawers, find your passport　before you can fill in this form.

But once you have, then you were able to send it back to us and we can now get hold of your information　and continue the implementation of booking your travel tickets.

So this is kind of similar to a how a callback works. And the whole reason why we use it is because there's　usually an event that requires a little bit of waiting.

So similar to how when we talked about async and await, we were waiting for the network to give us the　data we need.

Well in this case we're actually waiting for the user interaction.

So we're already pre-specifying well this is what should happen when the user taps on the checkbox and　we're waiting for that event to actually happen in real life before we trigger that callback. So I hope　that cleared up callbacks a little bit for you and it might be a good idea to have a play around with　the code as it is at the moment　just to be able to see how it works, what you can do with callbacks and understand how we're able to　lift our state up and pass the information upstream through the use of callbacks.

So in the next lesson, we have a challenge for you to implement lifting state up and managing state yourself.

So for all of that and more, II'll see on the next lesson.