**SwiftUI Animation**

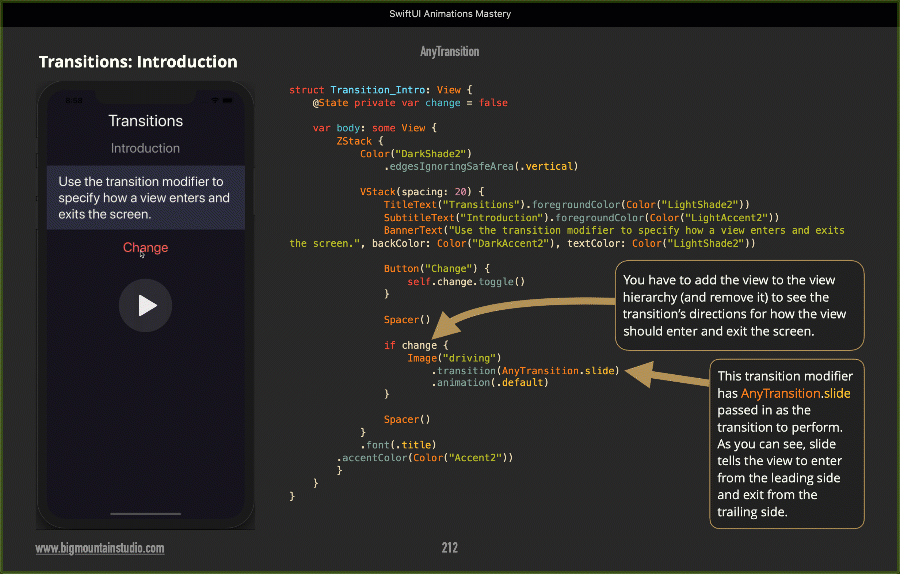
So, every view with a transition needs 3 things:

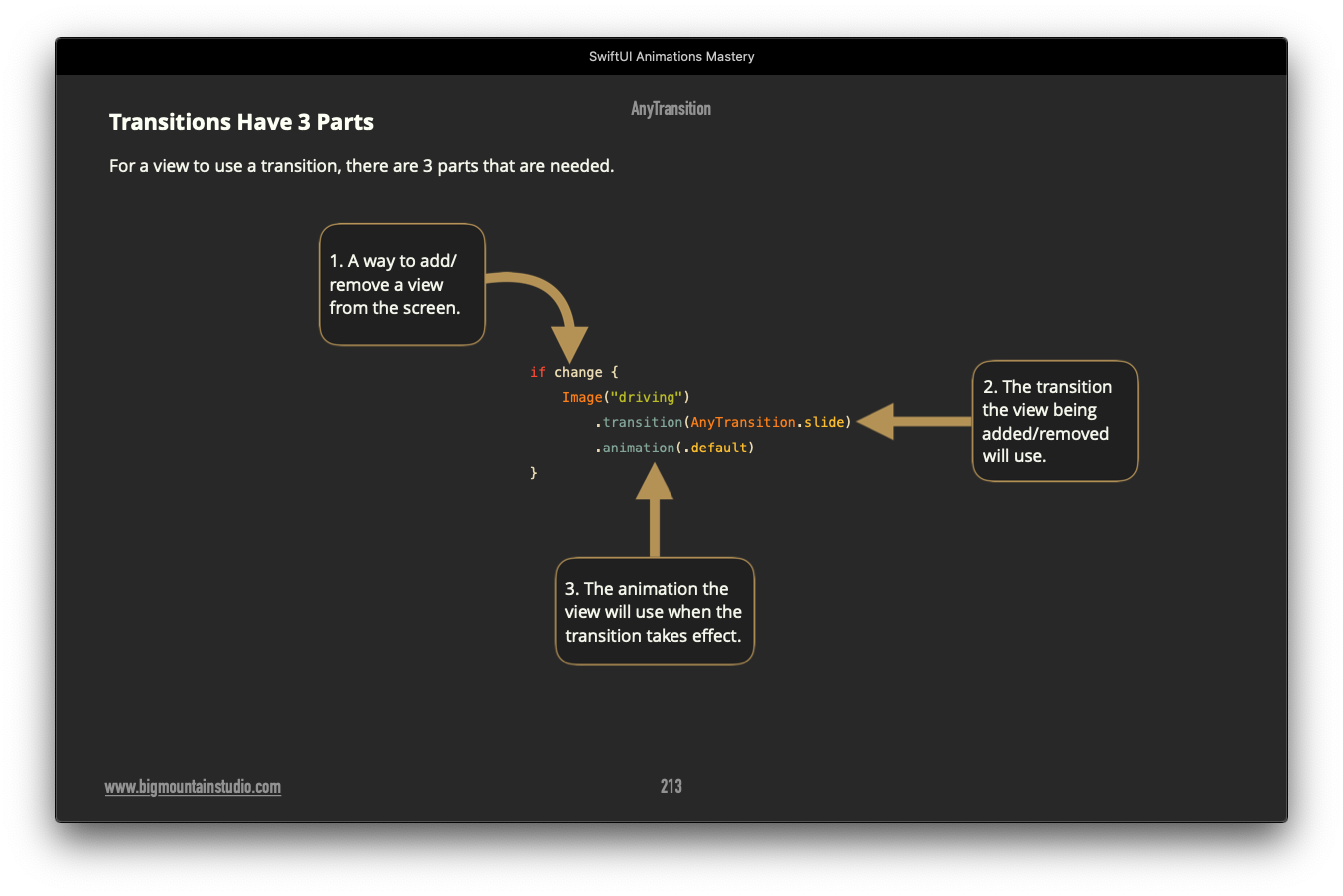
1. The view needs to be added or removed in some way. This is usually done with an if statement.
2. The view being added or removed needs the transition modifier applied to it.
3. There needs to be an animation modifier the transition can use. This animation can be applied either:

\* Directly to the view

\* To the view’s parent

\* To the variable that’s controlling the condition of when the view is inserted or removed (called an “explicit animation”)





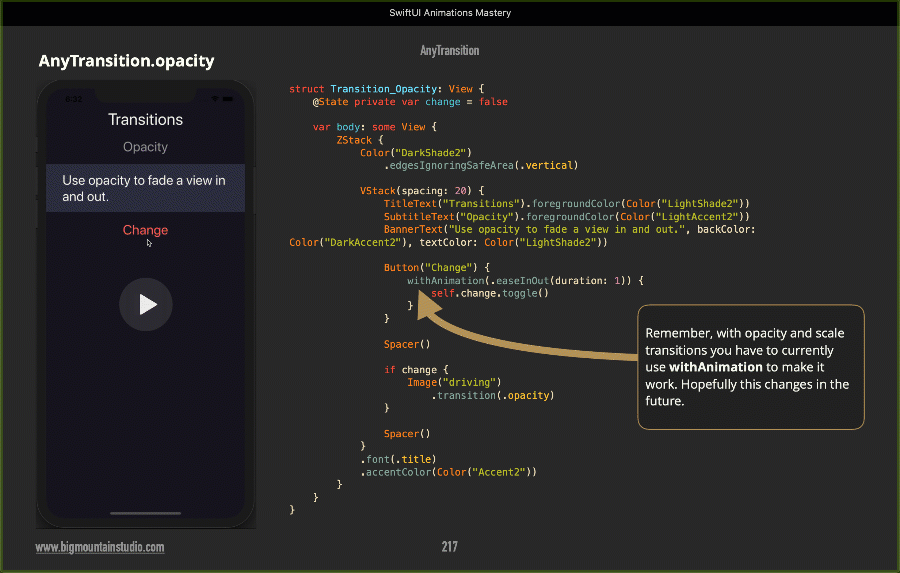
There are 5 transitions available in SwiftUI:

1. Slide - slide in the view from the leading side and remove it towards the trailing side.
2. Move - slide in from the direction specified and remove it from the same direction.
3. Opacity - fade in the view when inserted and fade out the view when removed.
4. Scale - insert the view FROM the scale specified and reverse the effect when removed.
5. Offset - like the move transition by you can set your own x and y coordinate.

**Working with Animations**

An animation is necessary to see the transition effect specified. You have three choices for applying an animation:

1. Apply an animation directly to the view (like in the example above)
2. Apply an animation to the parent view
3. Use an explicit animation (withAnimation) around the variable change that controls when the view is inserted/removed.



Combining Animations

You can combine these transition effects to work together using the combine function.

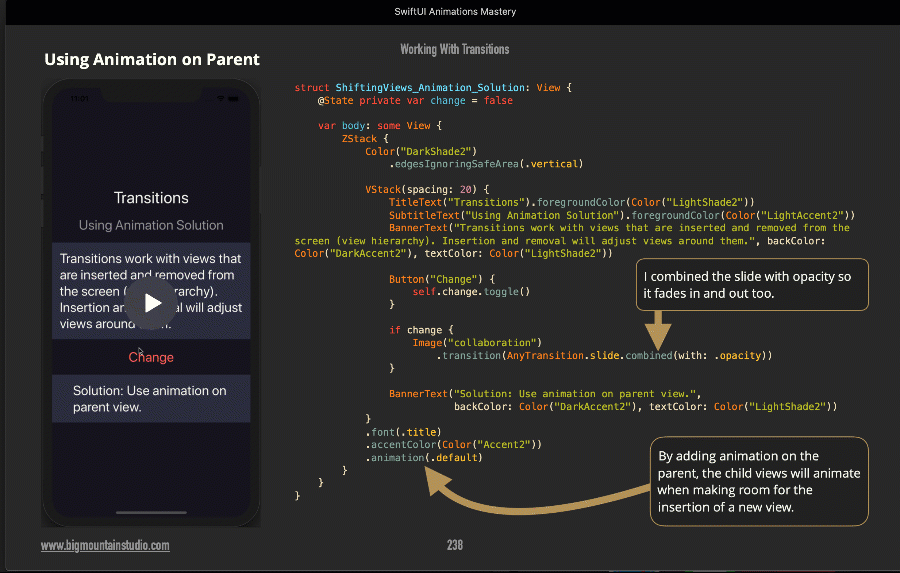
Example:

if change {

Image("window")

.transition(AnyTransition.slide.combined(with: .opacity))

}



**Create Different Insertion & Removal Effects**

So far, you have seen how transitions insert and remove using the same effect. But you can use another modifier that lets you specify a different removal effect than the insertion effect.

For example, you can insert the view with a slide effect and then remove it with a move effect.

You accomplish this by using the asymmetric function. The word “symmetric” means “exactly the same”. And as you can guess, "asymmetric" means "not the same". Not the same insertion and removal effect to be precise in this case.

Here is an example of how this works:

**.transition(AnyTransition.asymmetric(insertion: .slide, removal: .move(edge: .bottom)))**

You may also want to use different animations for insertion and removal. To achieve this, use explicit animations (withAnimation) when you change the variable that controls the insertion and removal.

