

UNDERSTANDING SWIFTUI'S STATE PROPERTY WRAPPERS

What's the difference between `@Binding`, `@State`, `@Environment`, `@StateObject`, `@ObservedObject` and `@EnvironmentObject`?

	The view creates and owns the state	The state is passed to the view's initializer	The state traverses the view tree
State has value semantics (aka a <code>struct</code>)	<code>@State</code>	<code>@Binding</code>	<code>@Environment</code>
State has reference semantics (aka a <code>class</code>)	<code>@StateObject</code>	<code>@ObservedObject</code>	<code>@EnvironmentObject</code>

How should you read this?

The table shows the intended usage for each property wrapper. (e.g. if the state you're passing around has value semantics and you want it to traverse the view tree, your best bet is `@Environment`.)

But X can be used with Y!

While nothing stops you from using, say, a reference semantics object with `@State`, you'll find it's useless since changes to your state won't trigger view updates.