@ViewBuilder is a Swift attribute that allows you to build views dynamically. It is typically used to create custom views that can combine other views to create their own complex layout.

@ViewBuilder works by taking a closure as an argument. The closure can return any number of views, including other TupleViews. SwiftUI will then combine the returned views to create a single view.

Here is an example of a custom view that uses @ViewBuilder to create a simple layout:

Swift

```swift
struct MyView: View {
    @ViewBuilder var content: () -> some View

    var body: some View {
        VStack {
            content()
        }
    }
}
```

This view can be used as follows:

Swift

```swift
struct ContentView: View {
    var body: some View {
        MyView {
            Text("Hello")
            Text("World")
        }
    }
}
```

When this code is rendered, it will display the text "Hello" and "World" on two separate lines.

@ViewBuilder is a powerful tool for creating reusable and composable views in SwiftUI. It allows you to create custom views that can combine other views to create their own complex layout.

Here are some of the benefits of using @ViewBuilder:

- It makes your code more concise and expressive. You can use @ViewBuilder to create complex view hierarchies without having to worry about the underlying implementation.
- It makes your code more reusable and composable. You can create custom views that can be used in different contexts and combined with other views to create new views.
- It makes your code more readable and maintainable. The code that uses @ViewBuilder is typically more readable and maintainable than the code that does not use it.

Overall, @ViewBuilder is a valuable tool for any SwiftUI developer. It allows you to create powerful and reusable views that can be used to build complex and beautiful user interfaces.