

[SwiftUI](#) / [ViewBuilder](#)

Structure

ViewBuilder

A custom parameter attribute that constructs views from closures.

iOS 13.0+ | iPadOS 13.0+ | Mac Catalyst 13.0+ | macOS 10.15+ | tvOS 13.0+ | visionOS 1.0+ | watchOS 6.0+

```
@resultBuilder  
struct ViewBuilder
```

Mentioned in

-  Declaring a custom view
-  Populating SwiftUI menus with adaptive controls

Overview

You typically use [ViewBuilder](#) as a parameter attribute for child view-producing closure parameters, allowing those closures to provide multiple child views. For example, the following `contextMenu` function accepts a closure that produces one or more views via the view builder.

```
func contextMenu<MenuItemType: View>(  
    @ViewBuilder menuItems: () -> MenuItemType  
) -> some View
```

Clients of this function can use multiple-statement closures to provide several child views, as shown in the following example:

```
myView.contextMenu {  
    Text("Cut")
```

```
Text("Copy")
Text("Paste")
if isSymbol {
    Text("Jump to Definition")
}
}
```

Topics

Building content

```
static func buildBlock() -> EmptyView
```

Builds an empty view from a block containing no statements.

```
static buildBlock(_:)
```

Passes a single view written as a child view through unmodified.

```
static func buildExpression<Content>(Content) -> Content
```

Builds an expression within the builder.

Conditionally building content

```
static func buildEither<TrueContent, FalseContent>(first: TrueContent)
-> _ConditionalContent<TrueContent, FalseContent>
```

Produces content for a conditional statement in a multi-statement closure when the condition is true.

```
static func buildEither<TrueContent, FalseContent>(second: FalseContent)
-> _ConditionalContent<TrueContent, FalseContent>
```

Produces content for a conditional statement in a multi-statement closure when the condition is false.

```
static func buildIf<Content>(Content?) -> Content?
```

Produces an optional view for conditional statements in multi-statement closures that's only visible when the condition evaluates to true.

```
static func buildLimitedAvailability<Content>(Content) -> AnyView
```

Processes view content for a conditional compiler-control statement that performs an availability check.

See Also

Creating a view

 Declaring a custom view

Define views and assemble them into a view hierarchy.

`protocol View`

A type that represents part of your app's user interface and provides modifiers that you use to configure views.