New fundamentals

Creating a view

SwiftUI Declaring a custom view

∨ Ⓟ **View**

   Implementing a custom view

   Ⓟ var body: Self.Body

   ☰ Body

   Ⓜ func modifier<T>(T) -> Modified...

› ☰ Previews in Xcode

   Configuring view elements

› ☰ Accessibility modifiers

› ☰ Appearance modifiers

› ☰ Text and symbol modifiers

› ☰ Auxiliary view modifiers

› ☰ Chart view modifiers

   Drawing views

› ☰ Style modifiers

› ☰ Layout modifiers

› ☰ Graphics and rendering modifiers

   Providing interactivity

› ☰ Input and event modifiers

🔍 Filter   /

› State modifiers

   Deprecated modifiers

› Deprecated modifiers

   Instance Methods

Instance Method

widgetAccentable(_:)

Adds the view and all of its subviews to the accented group.

iOS 16.0+ | iPadOS 16.0+ | Mac Catalyst 16.0+ | macOS 13.0+ | visionOS 26.0+ | watchOS 9.0+

```
@MainActor @preconcurrency
func widgetAccentable(_ accentable: Bool = true) -> some View
```

Parameters

accentable

A Boolean value that indicates whether to add the view and its subviews to the accented group.

Discussion

When the system renders the widget using the `WidgetKit/WidgetRendering Mode/accented` mode, it divides the widget's view hierarchy into two groups: the accented group and the default group. It then applies a different color to each group.

When applying the colors, the system treats the widget's views as if they were template images. It ignores the view's color — rendering the new colors based on the view's alpha channel.

To control your view's appearance, add the `widgetAccentable(_:)` modifier to part of your view's hierarchy. If the `accentable` parameter is `true`, the system adds that view and all of its subviews to the accent group. It puts all other views in the default group.

```
var body: some View {
    VStack {
        Text("MON")
            .font(.caption)
            .widgetAccentable()
        Text("6")
            .font(.title)
    }
}
```

> **Important**
>
> After you call `widgetAccentable(true)` on a view moving it into the accented group, calling `widgetAccentable(false)` on its subviews doesn't move the subviews back into the default group.