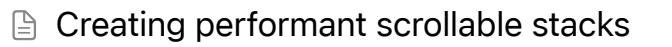SwiftUI / ScrollView

Structure

ScrollView

A scrollable view.

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

```
struct ScrollView<Content> where Content : View
```

Mentioned in

📄 Picking container views for your content

📄 Creating performant scrollable stacks

📄 Grouping data with lazy stack views

Overview

The scroll view displays its content within the scrollable content region. As the user performs platform-appropriate scroll gestures, the scroll view adjusts what portion of the underlying content is visible. `ScrollView` can scroll horizontally, vertically, or both, but does not provide zooming functionality.

In the following example, a `ScrollView` allows the user to scroll through a VStack containing 100 Text views. The image after the listing shows the scroll view's temporarily visible scrollbar at the right; you can disable it with the `showsIndicators` parameter of the `ScrollView` initializer.

```
var body: some View {
    ScrollView {
        VStack(alignment: .leading) {
            ForEach(0..<100) {
                Text("Row \($0)")
```

```
            }
        }
    }
}
```

```
Row 0
Row 1
Row 2
Row 3
Row 4
Row 5
Row 6
Row 7
Row 8
Row 9
Row 10
Row 11
Row 12
Row 13
Row 14
Row 15
Row 16
Row 17
```

Controlling Scroll Position

You can influence where a scroll view is initially scrolled by using the `defaultScroll Anchor(_:)` view modifier.

Provide a value of `UnitPoint/center`` to have the scroll view start in the center of its content when a scroll view is scrollable in both axes.

```
ScrollView([.horizontal, .vertical]) {
    // initially centered content
}
.defaultScrollAnchor(.center)
```

Or provide an alignment of `UnitPoint/bottom`` to have the scroll view start at the bottom of its content when a scroll view is scrollable in its vertical axes.

```
ScrollView {
    // initially bottom aligned content
}
.defaultScrollAnchor(.bottom)
```

After the scroll view initially renders, the user may scroll the content of the scroll view.

To perform programmatic scrolling, wrap one or more scroll views with a `ScrollViewReader`.

Topics

Creating a scroll view

~~init(Axis.Set, showsIndicators: Bool, content: () -> Content)~~

Creates a new instance that's scrollable in the direction of the given axis and can show indicators while scrolling.

`Deprecated`

`init(Axis.Set, content: () -> Content)`

Creates a new instance that's scrollable in the direction of the given axis and can show indicators while scrolling.

Configuring a scroll view

`var content: Content`

The scroll view's content.

`var axes: Axis.Set`

The scrollable axes of the scroll view.

`var showsIndicators: Bool`

A value that indicates whether the scroll view displays the scrollable component of the content offset, in a way that's suitable for the platform.

Supporting types

`var body: some View`

The content and behavior of the scroll view.

Relationships

Conforms To

View

See Also

Creating a scroll view

struct **ScrollViewReader**

A view that provides programmatic scrolling, by working with a proxy to scroll to known child views.

struct **ScrollViewProxy**

A proxy value that supports programmatic scrolling of the scrollable views within a view hierarchy.