

[SwiftUI](#) / VStack

Structure






VStack

A view that arranges its subviews in a vertical line.

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

```
@frozen
struct VStack<Content> where Content : View
```

Mentioned in

-  Building layouts with stack views
-  Creating performant scrollable stacks
-  Adding a background to your view
-  Inspecting view layout
-  Picking container views for your content

Overview

Unlike [LazyVStack](#), which only renders the views when your app needs to display them, a `VStack` renders the views all at once, regardless of whether they are on- or offscreen. Use the regular `VStack` when you have a small number of subviews or don't want the delayed rendering behavior of the "lazy" version.

The following example shows a simple vertical stack of 10 text views:

```
var body: some View {
    VStack(
        alignment: .leading,
```

```

        spacing: 10
    ) {
        ForEach(
            1...10,
            id: \.self
        ) {
            Text("Item \($0)")
        }
    }
}

```

Item 1

Item 2

Item 3

Item 4

Item 5

Item 6

Item 7

Item 8

Item 9

Item 10

Note

If you need a vertical stack that conforms to the [Layout](#) protocol, like when you want to create a conditional layout using [AnyLayout](#), use [VStackLayout](#) instead.

Topics

Creating a stack

```

init(alignment: HorizontalAlignment, spacing: CGFloat?, content: () ->
Content)

```

Creates an instance with the given spacing and horizontal alignment.

Relationships

Conforms To

View

See Also

Statically arranging views in one dimension



Building layouts with stack views

Compose complex layouts from primitive container views.

`struct HStack`

A view that arranges its subviews in a horizontal line.