Documentation    Language: Swift

SwiftUI / TextEditor

Structure

# TextEditor

A view that can display and edit long-form text.

iOS 14.0+ | iPadOS 14.0+ | Mac Catalyst 14.0+ | macOS 11.0+ | visionOS 1.0+

```
struct TextEditor
```

## Overview

A text editor view allows you to display and edit multiline, scrollable text in your app’s user interface. By default, the text editor view styles the text using characteristics inherited from the environment, like `font(_:)`, `foregroundColor(_:)`, and `multilineTextAlignment(_:)`.

You create a text editor by adding a `TextEditor` instance to the body of your view, and initialize it by passing in a `Binding` to a string variable in your app:

```
struct TextEditingView: View {
    @State private var fullText: String = "This is some editable text..."

    var body: some View {
        TextEditor(text: $fullText)
    }
}
```

To style the text, use the standard view modifiers to configure a system font, set a custom font, or change the color of the view’s text.

In this example, the view renders the editor’s text in gray with a custom font:

```swift
struct TextEditingView: View {
    @State private var fullText: String = "This is some editable text..."

    var body: some View {
        TextEditor(text: $fullText)
            .foregroundColor(Color.gray)
            .font(.custom("HelveticaNeue", size: 13))
    }
}
```

If you want to change the spacing or font scaling aspects of the text, you can use modifiers like `lineLimit(_:)`, `lineSpacing(_:)`, and `minimumScaleFactor(_:)` to configure how the view displays text depending on the space constraints. For example, here the `lineSpacing(_:)` modifier sets the spacing between lines to 5 points:

```swift
struct TextEditingView: View {
    @State private var fullText: String = "This is some editable text..."

    var body: some View {
        TextEditor(text: $fullText)
            .foregroundColor(Color.gray)
            .font(.custom("HelveticaNeue", size: 13))
            .lineSpacing(5)
    }
}
```

# Topics

## Creating a text editor

`init(text: Binding<String>)`

Creates a plain text editor.

## Initializers

`init(text:selection:)`

Creates a styled text editor.

# Relationships

## Conforms To

`View`

# See Also

## Getting text input

{} Building rich SwiftUI text experiences

Build an editor for formatted text using SwiftUI text editor views and attributed strings.

`struct TextField`

A control that displays an editable text interface.

`func textFieldStyle<S>(S) -> some View`

Sets the style for text fields within this view.

`struct SecureField`

A control into which people securely enter private text.