Text Input and Output

Display formatted text and get text input from the user.

Technology

SwiftUI

On This Page

Overview ⌄

Topics ⌄

See Also ⌄

Overview

To display read-only text, or read-only text paired with an image, use the built-in `Text` or `Label` views, respectively. Add modifiers to control the text's font, selectability, alignment, layout direction, and so on. When you need to collect text input from the user, use an appropriate text input view, like `TextField` or `TextEditor`.

To learn about the set of modifiers that you use to configure text that appears in a view, see Text and Symbol Modifiers.

Topics

Text Display

`struct Text`

A view that displays one or more lines of read-only text.

`struct Label`

A standard label for user interface items, consisting of an icon with a title.

`protocol TextSelectability`

A type that describes the ability to select text.

Text Entry

`struct TextField`

`struct SecureField`

A control into which the user securely enters private text.

`struct TextEditor`

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

Documentation › SwiftUI › Text Input and Output   Language: Swift   API Changes: Show ›

struct `TextInputAutocapitalization`

The kind of autocapitalization behavior applied during text input.

Text Style

📄 Applying Custom Fonts to Text

Add and use a font in your app that scales with Dynamic Type.

struct `Font`

An environment-dependent font.

struct `ScaledMetric`

A dynamic property that scales a numeric value.

enum `Case`

A scheme for transforming the capitalization of characters within text.

struct `DateStyle`

A predefined style used to display a `Date`.

struct `LineStyle`

Description of the style used to draw the line for `StrikethroughStyleAttribute` and `UnderlineStyleAttribute`.

enum `LegibilityWeight`

The Accessibility Bold Text user setting options.

enum `DynamicTypeSize`

A Dynamic Type size, which specifies how large scalable content should be.

struct `RedactionReasons`

The reasons to apply a redaction to data displayed on screen.

Text Layout

enum `TextAlignment`

An alignment position for text along the horizontal axis.

enum `TruncationMode`

The type of truncation to apply to a line of text when it's too long to fit in the available space.

enum `LayoutDirection`

Localization

📄 Preparing Views for Localization

Specify hints and add strings to localize your SwiftUI views.

struct LocalizedStringKey

The key used to look up an entry in a strings file or strings dictionary file.

Deprecated Types

enum ~~ContentSizeCategory~~

The sizes that you can specify for content.

Deprecated

See Also

User Interface Elements

▤ View Fundamentals

Create a hierarchy of configured views.

▤ Images

Add images to your user interface.

▤ Controls and Indicators

Display values and get user selections.

▤ Shapes

Trace and fill built-in and custom shapes with a color, gradient, or other pattern.

▤ Drawing and Graphics

Enhance your views with graphical effects and customized drawings.