Structure

TextSelection

Represents a selection of text.

iOS 18.0+ | iPadOS 18.0+ | Mac Catalyst 18.0+ | macOS 15.0+ | tvOS 18.0+ | visionOS 2.0+ | watchOS 11.0+

```
struct TextSelection
```

Overview

A selection is either an insertion point (e.g. a cursor in the text), a selection over a range of text or on macOS, multiple selections.

This is frequently used to represent selection of text in a `TextField` or `TextEditor`. The following example shows a text editor that leverages text selection to offer live suggestions based on the current selection.

```
struct SuggestionTextEditor: View {
    @State var text: String = ""
    @State var selection: TextSelection? = nil

    var body: some View {
        VStack {
            TextEditor(text: $text, selection: $selection)
            // A helper view that offers live suggestions based on selection.
            SuggestionsView(
                substrings: getSubstrings(text: text, indices: selection?.indices))
        }
    }

    private func getSubstrings(
        text: String, indices: TextSelection.Indices?
    ) -> [Substring] {
        // Resolve substrings representing the current selection...
    }
}

struct SuggestionsView: View { ... }
```

You can also use the `textSelectionAffinity(_:)` modifier to specify a selection affinity on the given hierarchy:

```
struct SuggestionTextEditor: View {
    @State var text: String = ""
    @State var selection: TextSelection? = nil

    var body: some View {
        VStack {
            TextEditor(text: $text, selection: $selection)
            // A helper view that offers live suggestions based on selection.
            SuggestionsView(
                substrings: getSubstrings(text: text, indices: selection?.indices))
        }
        .textSelectionAffinity(.upstream)
    }

    private func getSubstrings(
        text: String, indices: TextSelection.Indices?
    ) -> [Substring] {
        // Resolve substrings representing the current selection...
    }
}

struct SuggestionsView: View { ... }
```

Topics

Initializers

`init(insertionPoint: String.Index)`

Create a selection at a given insertion point.

`init(range: Range<String.Index>)`

Create a single selection with a given range.

`init(ranges: RangeSet<String.Index>)`

Create multiple selections with a given range-set.

Instance Properties

`var affinity: TextSelectionAffinity`

Return the selection affinity of the selection.

`var indices: TextSelection.Indices`

Return the current text selection indices.

`var isInsertion: Bool`

Return `true` if the selection is an insertion point.

Enumerations

`enum Indices`

The indices of the current selection.

Relationships

Conforms To

`Equatable`, `Hashable`

See Also

Selecting text

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

Controls whether people can select text within this view.

```
protocol TextSelectability
```

A type that describes the ability to select text.

```
func textSelectionAffinity(TextSelectionAffinity) -> some View
```

Sets the direction of a selection or cursor relative to a text character.

```
var textSelectionAffinity: TextSelectionAffinity
```

A representation of the direction or association of a selection or cursor relative to a text character. This concept becomes much more prominent when dealing with bidirectional text (text that contains both LTR and RTL scripts, like English and Arabic combined).

```
enum TextSelectionAffinity
```

A representation of the direction or association of a selection or cursor relative to a text character. This concept becomes much more prominent when dealing with bidirectional text (text that contains both LTR and RTL scripts, like English and Arabic combined).

```
struct AttributedTextSelection
```

Represents a selection of attributed text.