

[SwiftUI](#) / Accessible descriptions

API Collection

Accessible descriptions

Describe interface elements to help people understand what they represent.

Overview

SwiftUI can often infer some information about your user interface elements, but you can use accessibility modifiers to provide even more information for users that need it.



For design guidance, see [Accessibility](#) in the Accessibility section of the Human Interface Guidelines.

Topics

Applying labels

```
func accessibilityLabel(_ :)
```

Adds a label to the view that describes its contents.

```
func accessibilityLabel(_:isEnabled:)
```

Adds a label to the view that describes its contents.

```
func accessibilityLabel<V>(content: (PlaceholderContentView<Self>) -> V  
) -> some View
```

Adds a label to the view that describes its contents.

```
func accessibilityInputLabels(_:)
```

Sets alternate input labels with which users identify a view.

```
func accessibilityInputLabels(_:isEnabled:)
```

Sets alternate input labels with which users identify a view.

```
func accessibilityLabeledPair<ID>(role: AccessibilityLabeledPairRole,  
id: ID, in: Namespace.ID) -> some View
```

Pairs an accessibility element representing a label with the element for the matching content.

```
enum AccessibilityLabeledPairRole
```

The role of an accessibility element in a label / content pair.

Describing values

```
func accessibilityValue(_:)
```

Adds a textual description of the value that the view contains.

```
func accessibilityValue(_:isEnabled:)
```

Adds a textual description of the value that the view contains.

Describing content

```
func accessibilityTextContentType(AccessibilityTextContentType) ->  
ModifiedContent<Self, AccessibilityAttachmentModifier>
```

Sets an accessibility text content type.

```
func accessibilityHeading(AccessibilityHeadingLevel) -> ModifiedContent  
<Self, AccessibilityAttachmentModifier>
```

Sets the accessibility level of this heading.

```
enum AccessibilityHeadingLevel
```

The hierarchy of a heading in relation to other headings.

```
struct AccessibilityTextContentType
```

Textual context that assistive technologies can use to improve the presentation of spoken text.

Describing charts

```
func accessibilityChartDescriptor<R>(R) -> some View
```

Adds a descriptor to a View that represents a chart to make the chart's contents accessible to all users.

```
protocol AXChartDescriptorRepresentable
```

A type to generate an `AXChartDescriptor` object that you use to provide information about a chart and its data for an accessible experience in VoiceOver or other assistive technologies.

Adding custom descriptions

```
func accessibilityCustomContent(_:_:importance:)
```

Add additional accessibility information to the view.

```
struct AccessibilityCustomContentKey
```

Key used to specify the identifier and label associated with an entry of additional accessibility information.

Assigning traits to content

```
func accessibilityAddTraits(AccessibilityTraits) -> ModifiedContent<Self, AccessibilityAttachmentModifier>
```

Adds the given traits to the view.

```
func accessibilityRemoveTraits(AccessibilityTraits) -> ModifiedContent<Self, AccessibilityAttachmentModifier>
```

Removes the given traits from this view.

```
struct AccessibilityTraits
```

A set of accessibility traits that describe how an element behaves.

Offering hints

```
func accessibilityHint(_:)
```

Communicates to the user what happens after performing the view's action.

```
func accessibilityHint(_:isEnabled:)
```

Communicates to the user what happens after performing the view's action.

Configuring VoiceOver

```
func speechAdjustedPitch(Double) -> some View
```

Raises or lowers the pitch of spoken text.

```
func speechAlwaysIncludesPunctuation(Bool) -> some View
```

Sets whether VoiceOver should always speak all punctuation in the text view.

```
func speechAnnouncementsQueued(Bool) -> some View
```

Controls whether to queue pending announcements behind existing speech rather than interrupting speech in progress.

```
func speech SpellsOutCharacters(Bool) -> some View
```

Sets whether VoiceOver should speak the contents of the text view character by character.

See Also

Accessibility

☰ Accessibility fundamentals

Make your SwiftUI apps accessible to everyone, including people with disabilities.

☰ Accessible appearance

Enhance the legibility of content in your app's interface.

☰ Accessible controls

Improve access to actions that your app can undertake.

☰ Accessible navigation

Enable users to navigate to specific user interface elements using rotors.