

[UIKit / UIStepper](#)

Class

UIStepper

A control for incrementing or decrementing a value.

iOS 5.0+ | iPadOS 5.0+ | Mac Catalyst 13.1+ | visionOS 1.0+

```
@MainActor  
class UIStepper
```

Mentioned in

- 📄 Attaching gesture recognizers to UIKit controls
- 📄 Choosing a user interface idiom for your Mac app

Overview

By default, pressing and holding a stepper's button increments or decrements the stepper's value repeatedly. The rate of change depends on how long the user continues pressing the control. To turn off this behavior, set the [autorepeat](#) property to [false](#).

The maximum value must be greater than or equal to the minimum value. If you set a maximum or minimum value that would break this invariant, both values are set to the new value. For example, if the minimum value is 200 and you set a maximum value of 100, then both the minimum and maximum become 200.

Important

[UIStepper](#) isn't available when the user interface idiom is [UIUserInterfaceIdiom.mac](#).

Topics

Configuring the stepper

`var isContinuous: Bool`

A Boolean value that determines whether to send value changes during user interaction or after user interaction ends.

`var autorepeat: Bool`

A Boolean value that determines whether to repeatedly change the stepper's value as the user presses and holds a stepper button.

`var wraps: Bool`

A Boolean value that determines whether the stepper can wrap its value to the minimum or maximum value when incrementing and decrementing the value.

`var minValue: Double`

The lowest possible numeric value for the stepper.

`var maxValue: Double`

The highest possible numeric value for the stepper.

`var stepValue: Double`

The step, or increment, value for the stepper.

Accessing the stepper's value

`var value: Double`

The numeric value of the stepper.

Customizing appearance

`func backgroundImage(for: UIControl.State) -> UIImage?`

Returns the background image associated with the specified control state.

`func setBackgroundImage(UIImage?, for: UIControl.State)`

Sets the background image for the control when it's in the specified state.

`func decrementImage(for: UIControl.State) -> UIImage?`

Returns the image used for the decrement glyph of the control.

```
func setDecrementImage(UIImage?, for: UIControl.State)
```

Sets the image to use for the decrement glyph of the control.

```
func dividerImage(forLeftSegmentState: UIControl.State, rightSegmentState: UIControl.State) -> UIImage?
```

Returns the divider image for the given combination of left and right states.

```
func setDividerImage(UIImage?, forLeftSegmentState: UIControl.State, rightSegmentState: UIControl.State)
```

Sets the image to use for the given combination of left and right states.

```
func incrementImage(for: UIControl.State) -> UIImage?
```

Returns the image used for the increment glyph of the control.

```
func setIncrementImage(UIImage?, for: UIControl.State)
```

Sets the image to use for the increment glyph of the control.

Relationships

Inherits From

UIControl

Conforms To

CALayerDelegate

CVarArg

CustomDebugStringConvertible

CustomStringConvertible

Equatable

Hashable

NSCoding

NSObjectProtocol

NSTouchBarProvider

Sendable

SendableMetatype

UIAccessibilityIdentification

UIActivityItemsConfigurationProviding

UIAppearance
UIAppearanceContainer
UIContextMenuInteractionDelegate
UICoordinateSpace
UIDynamicItem
 UIFocusEnvironment
 UIFocusItem
 UIFocusItemContainer
 UILargeContentViewerItem
 UIPasteConfigurationSupporting
 UIPopoverPresentationControllerSourceItem
 UIResponderStandardEditActions
 UITraitChangeObservable
 UITraitEnvironment
 UIUserActivityRestoring

See Also

Controls

- 📄 Responding to control-based events using target-action
 - Handle user input by connecting buttons, sliders, and other controls to your app's code using the target-action design pattern.

`class UIControl`

The base class for controls, which are visual elements that convey a specific action or intention in response to user interactions.

`class UIButton`

A control that executes your custom code in response to user interactions.

`class UIColorWell`

A control that displays a color picker.

`class UIDatePicker`

A control for inputting date and time values.

`class UIPageControl`

A control that displays a horizontal series of dots, each of which corresponds to a page in the app's document or other data-model entity.

`class UISegmentedControl`

A horizontal control that consists of multiple segments, each segment functioning as a discrete button.

`class UISlider`

A control for selecting a single value from a continuous range of values.

`class UISwitch`

A control that offers a binary choice, such as on/off.