

Documentation

< All Technologies

SwiftUI

View

Implementing a custom view

var body: Self.Body

Body

func modifier<T>(T) -> Modified...

> Previews in Xcode

Configuring view elements

> Accessibility modifiers

> Appearance modifiers

> Text and symbol modifiers

> Auxiliary view modifiers

> Chart view modifiers

Drawing views

> Style modifiers

> Layout modifiers

> Graphics and rendering modifiers

Providing interactivity

> Input and event modifiers

Filter

/

SwiftUI / View / padding(_:_:)

Instance Method

padding(_:_:)

Adds an equal padding amount to specific edges of this view.

iOS 13.0+ | iPadOS 13.0+ | Mac Catalyst 13.0+ | macOS 10.15+ | tvOS 13.0+ |

visionOS 1.0+ | watchOS 6.0+

```
nonisolated
func padding(
    _ edges: Edge.Set = .all,
    _ length: CGFloat? = nil
) -> some View
```

Parameters

edges

The set of edges to pad for this view. The default is `all`.

length

An amount, given in points, to pad this view on the specified edges. If you set the value to `nil`, SwiftUI uses a platform-specific default amount. The default value of this parameter is `nil`.

Return Value

A view that's padded by the specified amount on the specified edges.

Mentioned in

Laying out a simple view

Discussion

Use this modifier to add a specified amount of padding to one or more edges of the view. Indicate the edges to pad by naming either a single value from `Edge.Set`, or by specifying an `OptionSet` that contains edge values:

```
VStack {
    Text("Text padded by 20 points on the bottom and trailing edges.")
        .padding([.bottom, .trailing], 20)
        .border(.gray)
    Text("Unpadded text for comparison.")
        .border(.yellow)
}
```

The order in which you apply modifiers matters. The example above applies the padding before applying the border to ensure that the border encompasses the padded region:

Text padded by 20 points on the bottom and trailing edges.

Unpadded text for comparison.

You can omit either or both of the parameters. If you omit the `length`, SwiftUI uses a default amount of padding. If you omit the `edges`, SwiftUI applies the padding to all edges. Omit both to add a default padding all the way around a view. SwiftUI chooses a default amount of padding that's appropriate for the platform and the presentation context.

```
VStack {
    Text("Text with default padding.")
        .padding()
        .border(.gray)
    Text("Unpadded text for comparison.")
        .border(.yellow)
}
```

The example above looks like this in iOS under typical conditions:

To control the amount of padding independently for each edge, use `padding(_:_:)`. To pad all outside edges of a view by a specified amount, use `padding(_:_:)`.

See Also

Adding padding around a view

func `padding(_:_:)`

Adds a different padding amount to each edge of this view.

func `padding3D(_:_:)`

Pads this view using the edge insets you specify.

func `padding3D(Edge3D.Set, CGFloat?) -> some View`

Pads this view using the edge insets you specify.

func `scenePadding(Edge.Set) -> some View`

Adds padding to the specified edges of this view using an amount that's appropriate for the current scene.

func `scenePadding(ScenePadding, edges: Edge.Set) -> some View`

Adds a specified kind of padding to the specified edges of this view using an amount that's appropriate for the current scene.

struct `ScenePadding`

The padding used to space a view from its containing scene.

Apple > Developer > Documentation

Platforms

iOS

iPadOS

macOS

tvOS

watchOS

Tools

Swift

SwiftUI

Swift Playground

TestFlight

Xcode

Xcode Cloud

SF Symbols

Topics & Technologies

Accessibility

Accessories

App Extension

App Store

App Store & V

Audio & Video

Augmented Reality

Design

Distribution

Education

Fonts

Games

Health & Fitness

In-App Purchase

In-App Review

Localization

Maps & Location

Machine Learning & AI

Open Source

Security

Safari & Web

Resources

Documentation

Tutorials

Downloads

Forums

Videos

Support

Support Articles

Contact Us

Bug Reporting

System Status

Account

Apple Developer

App Store Connect

Certificates & JKS & Profiles

Feedback Assistant

Feedback Assistant

Programs

Apple Developer Program

Apple Developer Enterprise Program

App Store Small Business Program

MFi Program

New Partner Program

Video Partner Program

Security Bounty Program

Security Research Device Program

Events

Meet with Apple

Meet with Apple

Apple Developer Centers

App Store Awards

Apple Design Awards

Apple Developer Academies

WWDC

To submit feedback on documentation, visit [Feedback Assistant](#).

Copyright © 2025 Apple Inc. All rights reserved.

Terms of Use

Privacy Policy

Agreements and Guidelines

Light

Dark

Auto