

[SwiftUI](#) / [View](#) / `modifier(_:)`

## Instance Method

# modifier(\_:)

Applies a modifier to a view and returns a new 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 modifier<T>(_ modifier: T) -> ModifiedContent<Self, T>
```

## Parameters

### `modifier`

The modifier to apply to this view.

## Mentioned in

 [Reducing view modifier maintenance](#)

## Discussion

Use this modifier to combine a [View](#) and a [ViewModifier](#), to create a new view. For example, if you create a view modifier for a new kind of caption with blue text surrounded by a rounded rectangle:

```
struct BorderedCaption: ViewModifier {
    func body(content: Content) -> some View {
        content
            .font(.caption2)
            .padding(10)
```

```

        .overlay(
            RoundedRectangle(cornerRadius: 15)
                .stroke(lineWidth: 1)
        )
        .foregroundColor(Color.blue)
    }
}

```

You can use `modifier(_ :)` to extend `View` to create new modifier for applying the Bordered Caption defined above:

```

extension View {
    func borderedCaption() -> some View {
        modifier(BorderedCaption())
    }
}

```

Then you can apply the bordered caption to any view:

```

Image(systemName: "bus")
    .resizable()
    .frame(width:50, height:50)
Text("Downtown Bus")
    .borderedCaption()

```



Downtown Bus

## See Also

### Modifying a view



Configuring views

Adjust the characteristics of a view by applying view modifiers.



Reducing view modifier maintenance

Bundle view modifiers that you regularly reuse into a custom view modifier.

`protocol ViewModifier`

A modifier that you apply to a view or another view modifier, producing a different version of the original value.

`struct EmptyModifier`

An empty, or identity, modifier, used during development to switch modifiers at compile time.

`struct ModifiedContent`

A value with a modifier applied to it.

`protocol EnvironmentalModifier`

A modifier that must resolve to a concrete modifier in an environment before use.

```
struct ManipulableModifier
```

```
struct ManipulableResponderModifier
```

```
struct ManipulableTransformBindingModifier
```

```
struct ManipulationGeometryModifier
```

```
struct ManipulationGestureModifier
```

```
struct ManipulationUsingGestureStateModifier
```

```
enum Manipulable
```

A namespace for various manipulable related types.