

[SwiftUI](#) / [View](#) / `contextMenu(menuItems:)`

Instance Method

contextMenu(menuItems:)

Adds a context menu to a view.

iOS 13.0+ | iPadOS 13.0+ | Mac Catalyst 13.0+ | macOS 10.15+ | tvOS 14.0+ | visionOS 1.0+ |

watchOS 6.0–7.0 **Deprecated**

```
nonisolated
func contextMenu<MenuItems>(@ViewBuilder menuItems: () -> Menu
Items) -> some View where MenuItems : View
```

Parameters

`menuItems`

A closure that produces the menu's contents. You can deactivate the context menu by returning nothing from the closure.

Return Value

A view that can display a context menu.

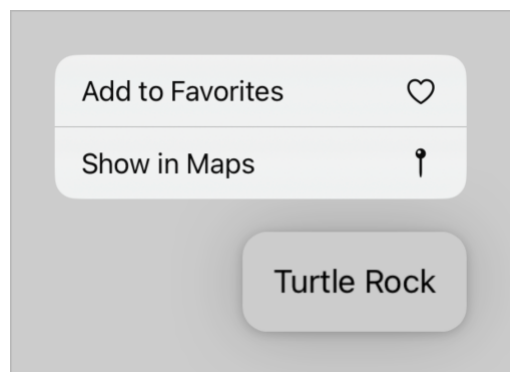
Discussion

Use this modifier to add a context menu to a view in your app's user interface. Compose the menu by returning controls like [Button](#), [Toggle](#), and [Picker](#) from the `menuItems` closure. You can also use [Menu](#) to define submenus or [Section](#) to group items.

The following example creates a [Text](#) view that has a context menu with two buttons:

```
Text("Turtle Rock")
    .padding()
    .contextMenu {
        Button {
            // Add this item to a list of favorites.
        } label: {
            Label("Add to Favorites", systemImage: "heart")
        }
        Button {
            // Open Maps and center it on this item.
        } label: {
            Label("Show in Maps", systemImage: "mappin")
        }
    }
}
```

When someone activates the context menu with an action like touch and hold in iOS or iPadOS, the system displays the menu next to the content:



The system dismisses the menu if someone makes a selection, or taps or clicks outside the menu.

To customize the default preview, apply a `contentShape(: :eoFill:)` with a `contextMenu Preview` kind. For example, you can change the preview's corner radius or use a nested view as the preview.

Note

This view modifier produces a context menu on macOS, but that platform doesn't display a preview.

If you want to show a different preview, you can use `contextMenu(menuItems:preview:)`. To add a context menu to a container that supports selection, like a `List` or a `Table`, and to distinguish between menu activation on a selection and activation in an empty area of the container, use `contextMenu(forSelectionType:menu:primaryAction:)`.

See Also

Creating context menus

```
func contextMenu<M, P>(menuItems: () -> M, preview: () -> P) -> some View
```

Adds a context menu with a custom preview to a view.

```
func contextMenu<I, M>(forSelectionType: I.Type, menu: (Set<I>) -> M, primaryAction: ((Set<I>) -> Void)? ) -> some View
```

Adds an item-based context menu to a view.