Instance Method

contextMenu(menuItems:preview:)

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

iOS 16.0+ | iPadOS 16.0+ | Mac Catalyst 16.0+ | macOS 13.0+ | tvOS 16.0+ | visionOS 1.0+

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

Parameters

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

`preview`
A view that the system displays along with the menu.

Return Value

A view that can display a context menu with a preview.

Discussion

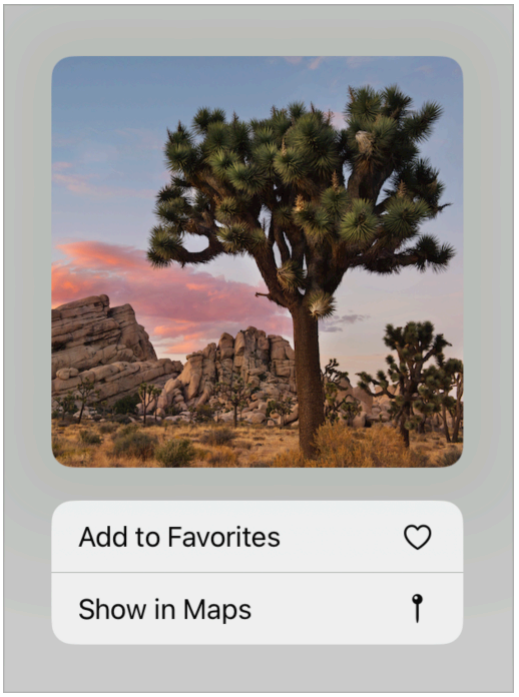
When you use this modifer to add a context menu to a view in your app's user interface, the system displays the custom preview beside the menu. Compose the menu by returning controls

like `Button`, `Toggle`, and `Picker` from the `menuItems` closure. You can also use `Menu` to define submenus or `Section` to group items.

Define the custom preview by returning a view from the `preview` closure. The system sizes the preview to match the size of its content. For example, you can add a two button context menu to a `Text` view, and use an `Image` as the preview:

```
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")
        }
    } preview: {
        Image("turtlerock") // Loads the image from an asset catalog.
    }
```

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



To customize the lift preview, shown while the system transitions to show your custom `preview`, apply a `contentShape(_:_:eoFill:)` with a `contextMenuPreview` kind. For example, you can change the lift preview's corner radius or use a nested view as the lift preview.

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

If you don't need a preview, use `contextMenu(menuItems:)` instead. If you want to add a context menu to a container that supports selection, like a `List` or a `Table`, and you want 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<MenuItems>(menuItems: () -> MenuItems) -> some View
```

Adds a context menu 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.