

[SwiftUI](#) / [EnvironmentValues](#) / `isFocused`

## Instance Property

# isFocused

Returns whether the nearest focusable ancestor has focus.

iOS 14.0+ | iPadOS 14.0+ | Mac Catalyst 14.0+ | macOS 11.0+ | tvOS 14.0+ | visionOS 1.0+ | watchOS 7.0+

```
var isFocused: Bool { get }
```

## Discussion

If there is no focusable ancestor, the value is `false`.

## See Also

### Managing focus state

```
func focused<Value>(FocusState<Value>.Binding, equals: Value) -> some View
```

Modifies this view by binding its focus state to the given state value.

```
func focused(FocusState<Bool>.Binding) -> some View
```

Modifies this view by binding its focus state to the given Boolean state value.

```
struct FocusState
```

A property wrapper type that can read and write a value that SwiftUI updates as the placement of focus within the scene changes.

```
struct FocusedValue
```

A property wrapper for observing values from the focused view or one of its ancestors.

`macro Entry()`

Creates an environment values, transaction, container values, or focused values entry.

`protocol FocusedValueKey`

A protocol for identifier types used when publishing and observing focused values.

`struct FocusedBinding`

A convenience property wrapper for observing and automatically unwrapping state bindings from the focused view or one of its ancestors.

`func searchFocused(FocusState<Bool>.Binding) -> some View`

Modifies this view by binding the focus state of the search field associated with the nearest searchable modifier to the given Boolean value.

`func searchFocused<V>(FocusState<V>.Binding, equals: V) -> some View`

Modifies this view by binding the focus state of the search field associated with the nearest searchable modifier to the given value.