

FREE TRIAL: Accelerate your app development career with Hacking with Swift+! >>

< What is the @EnvironmentObject property wrapper? What is the @Binding property wrapper? >

What is the @Environment property wrapper?

Paul Hudson @twostraws February 9th 2021

Updated for Xcode 13.2

SwiftUI gives us both **@Environment** and **@EnvironmentObject** property wrappers, but they are subtly different: whereas **@EnvironmentObject** allows us to inject arbitrary values into the environment, **@Environment** is specifically there to work with SwiftUI's own pre-defined keys.

For example, **@Environment** is great for reading out things like a Core Data managed object context, whether the device is in dark mode or light mode, what size class your view is being rendered with, and more – fixed properties that come from the system. In code, it looks like this:

```
@Environment(\.horizontalSizeClass) var horizontalSizeClass
@Environment(\.managedObjectContext) var managedObjectContext
```

On the other hand, **@EnvironmentObject** is designed for arbitrary objects to be read from the environment, like this:


```
@EnvironmentObject var order: Order
```

That difference might sound small, but it's important because of the way **@EnvironmentObject** is implemented. When we say that **order** is of type **Order** SwiftUI will look through its environment to find an object of that type and attach it to the **order** property. However, when using **@Environment** the same behavior isn't possible, because many things might share the same data type.

For example:

```
@Environment(\.accessibilityReduceMotion) var reduceMotion
@Environment(\.accessibilityReduceTransparency) var reduceTransparency
@Environment(\.accessibilityEnabled) var accessibilityEnabled
```

All three of those environment keys return a Boolean, so without specifying exactly which key we mean it would be impossible to read them correctly.



Raycast

SPONSORED Get work done faster with [Raycast](#). The Mac app brings the macOS Spotlight experience to the next level: Search files, create GitHub pull requests, close Jira issues and so much more. Use the [Xcode extension](#) to open recent projects, search iOS documentation and clear derived data. Or, build your own extension with an easy-to-use API.

[Find out more](#)

Sponsor Hacking with Swift and reach the world's largest Swift community!

Similar solutions...

- Observable objects, environment objects, and @Published
- All SwiftUI property wrappers explained and compared
- What is the @GestureState property wrapper?
- What is the @Published property wrapper?
- What is the @ScaledMetric property wrapper?

< What is the @EnvironmentObject property wrapper? What is the @Binding property wrapper? >



Was this page useful? Let us know!



Average rating: 4.5/5

[Click here to visit the Hacking with Swift store >>](#)



@twostraws



paul@hackingwithswift.com



Sponsor the site

[About](#)

[Glossary](#)

[Privacy Policy](#)

[Refund Policy](#)

[Update Policy](#)

[Code of Conduct](#)

Thanks for your support, roblack!

Swift, SwiftUI, the Swift logo, Swift Playgrounds, Xcode, Instruments, Cocoa Touch, Touch ID, AirDrop, iBeacon, iPhone, iPad, Safari, App Store, watchOS, tvOS, Mac and macOS are trademarks of Apple Inc., registered in the U.S. and other countries. Pulp Fiction is copyright © 1994 Miramax Films. Hacking with Swift is ©2021 Hudson Heavy Industries.



You are not logged in

[Log in or create account](#)