Moonshot

Which of these statements are true?

Click on any answer to show more detail.

1. **Correct**

   > **Option 1:** `GeometryReader` lets us read the size of a view's container.
   >
   > **Option 2:** Before trying to stretch the contents of an image view, we should use `aspectRatio(contentMode: .resize)`.
   >
   > You selected Option 1.

2. **Correct**

   > **Option 1:** If we specify the width of an image, we must also specify its height.
   >
   > **Option 2:** Higher layout priority values mean views are more likely to be allocated space in their container.
   >
   > You selected Option 2.

3. **Correct**

   > **Option 1:** Codable is capable of working with structs within structs as long as our Swift code matches the source data.
   >
   > **Option 2:** We can make a struct conform to one protocol, but not two at the same time.
   >
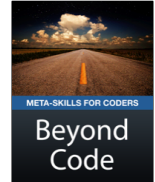   > You selected Option 1.

4. **Correct**

   > **Option 1:** `aspectRatio(contentMode: .fit)` is the same as `scaledToFit()`.
   >
   > **Option 2:** Scroll views can be horizontal or vertical, but not both.

> You selected Option 1.

5. **Correct**

> **Option 1:** All views have a layout priority of 0 by default.
>
> **Option 2:** SwiftUI view previews shouldn't have properties of their own.
>
> You selected Option 1.

6. **Correct**

> **Option 1:** The `first(where:)` method returns the array index of the first item that matches a predicate.
>
> **Option 2:** We can control the way SwiftUI highlights navigation links using the `buttonStyle()` modifier.
>
> You selected Option 2.

7. **Correct**

> **Option 1:** `NavigationLink` requires a `NavigationView` to work.
>
> **Option 2:** We can make a scroll view take up all available screen width by using `frame(maxWidth: .fill)`.
>
> You selected Option 1.

8. **Correct**

> **Option 1:** You can take all the text you want from Wikipedia; it's public domain.
>
> **Option 2:** The `.fill` content mode might mean parts of an image lie outside its container's frame.
>
> You selected Option 2.

9. **Correct**

Option 1: `NavigationView` lets us push a new custom view, or a basic type such as `Text`.

Option 2: `sheet()` requires a `NavigationView` to work.

You selected Option 1.

10. **Correct**

Option 1: Generics let us write code that can use a variety of different types.

Option 2: The `dateFormat` and `dateStyle` properties of `DateFormatter` do the same thing.

You selected Option 1.

11. **Correct**

Option 1: Angle brackets are [and].

Option 2: A nested struct is one that is placed inside another struct.

You selected Option 2.

12. **Correct**

Option 1: If Codable sees an optional property, it will only try to unarchive it if it exists in the source data.

Option 2: We can use `Spacer(minHeight:)` to force a spacer to be at least a certain height.

You selected Option 1.

Total score: 12/12

Back to Review menu

Note: if you're following the 100 Days of Swift or the 100 Days of SwiftUI, just close this window and return to where you were.

@twostraws

paul@hackingwithswift.com

About   Glossary   Privacy Policy   Refund Policy   Update Policy