

Accessibility Debugging in Xcode 11

What is Accessibility?

What is Accessibility?

VoiceOver

AssistiveTouch

Dynamic Type

Guided Access

Magnifier

Voice Control

Captioning

Increase Contrast

Reading Support

Zoom

Differentiate Color

Color Filters

What is Accessibility?

VoiceOver

AssistiveTouch

Dynamic Type

Guided Access

Magnifier

Voice Control

Captioning

Increase Contrast

Reading Support

Zoom

Differentiate Color

Color Filters

What is Accessibility?

VoiceOver

AssistiveTouch

Dynamic Type

Guided Access

Magnifier

Voice Control

Captioning

Increase Contrast

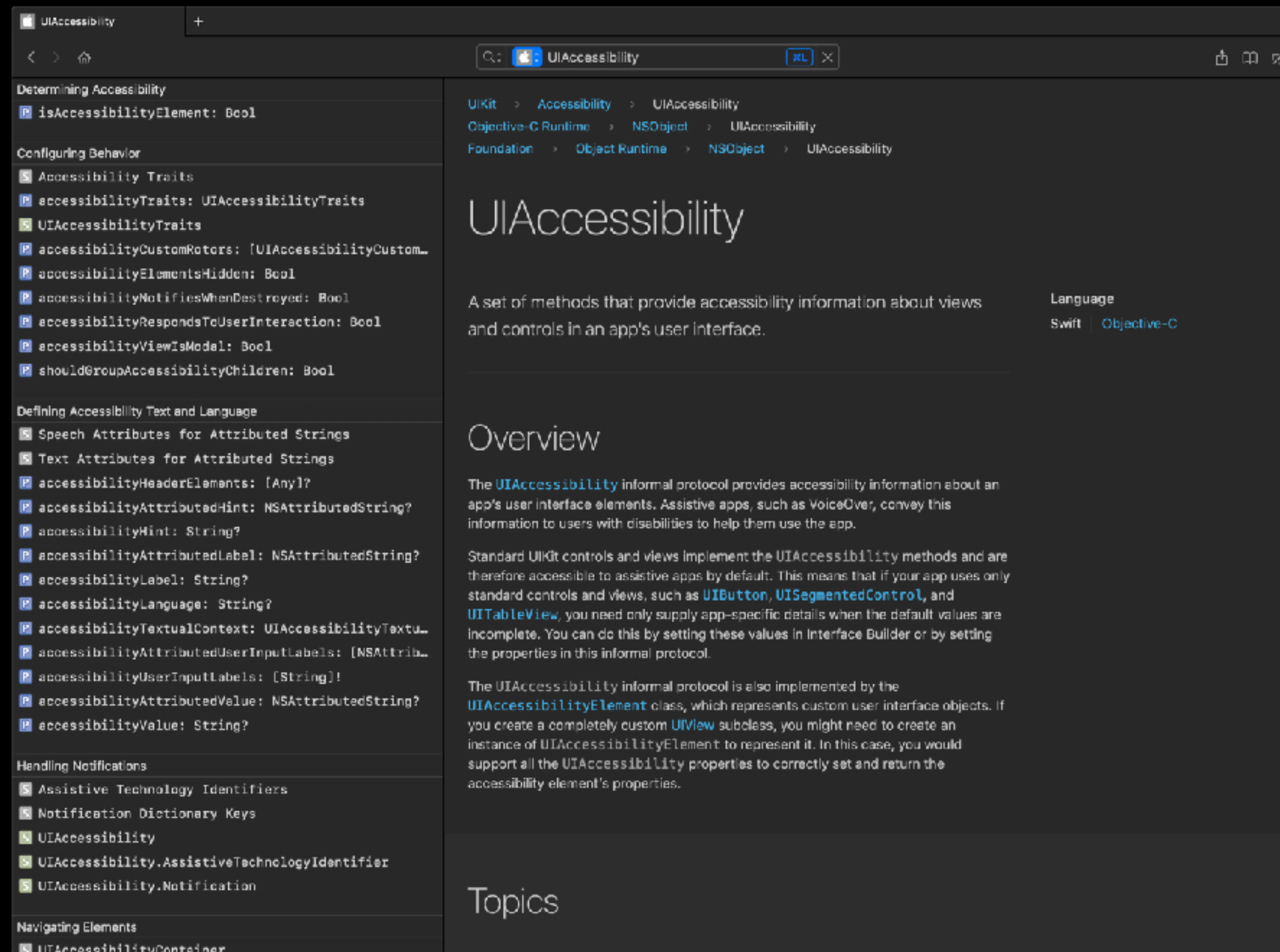
Reading Support

Zoom

Differentiate Color

Color Filters

What is Accessibility?



The screenshot shows the Xcode documentation for the `UIAccessibility` framework. The interface is split into two main sections: a left sidebar and a main content area.

Left Sidebar: This section lists various categories and properties related to accessibility. The categories include:

- Determining Accessibility:** `isAccessibilityElement: Bool`
- Configuring Behavior:** `Accessibility Traits`, `accessibilityTraits: UIAccessibilityTraits`, `UIAccessibilityTraits`, `accessibilityCustomRotors: [UIAccessibilityCustom...]`, `accessibilityElementsHidden: Bool`, `accessibilityNotifiesWhenDestroyed: Bool`, `accessibilityRespondsToUserInteraction: Bool`, `accessibilityViewIsModal: Bool`, and `shouldGroupAccessibilityChildren: Bool`.
- Defining Accessibility Text and Language:** `Speech Attributes for Attributed Strings`, `Text Attributes for Attributed Strings`, `accessibilityHeaderElements: [Any]?`, `accessibilityAttributedHint: NSAttributedString?`, `accessibilityHint: String?`, `accessibilityAttributedLabel: NSAttributedString?`, `accessibilityLabel: String?`, `accessibilityLanguage: String?`, `accessibilityTextualContext: UIAccessibilityTextu...`, `accessibilityAttributedUserInputLabels: [NSAttrib...`, `accessibilityUserInputLabels: [String]!`, `accessibilityAttributedValue: NSAttributedString?`, and `accessibilityValue: String?`.
- Handling Notifications:** `Assistive Technology Identifiers`, `Notification Dictionary Keys`, `UIAccessibility`, `UIAccessibility.AssistiveTechnologyIdentifier`, and `UIAccessibility.Notification`.
- Navigating Elements:** `UIAccessibilityContainer`.

Main Content Area: This section displays the title `UIAccessibility` and provides a brief description: "A set of methods that provide accessibility information about views and controls in an app's user interface." Below this, there is an "Overview" section that explains the `UIAccessibility` informal protocol and its implementation by standard UIKit controls and the `UIAccessibilityElement` class. A "Language" selector is visible, showing options for Swift and Objective-C.

Why Do Accessibility?

"[...] there's nothing on the iPhone or the iPad that you can do, that I can't do."

— Stevie Wonder

Why Do Accessibility?

“The accessibility support is as much a part of your user interface as the artwork is.”

— Apple

Testing Accessibility

Used to be painful

**Many trips between your app and
Settings**

Slow test/fix/verify cycle

Accessibility Inspector

Simulat...S 13.4 > All processes

Quicklook

Sign In, Button

Basic Hide

Label Sign In

Value <nil>

Traits Button

Identifier <nil>

Actions Hide

Activate Perform

Element Hide

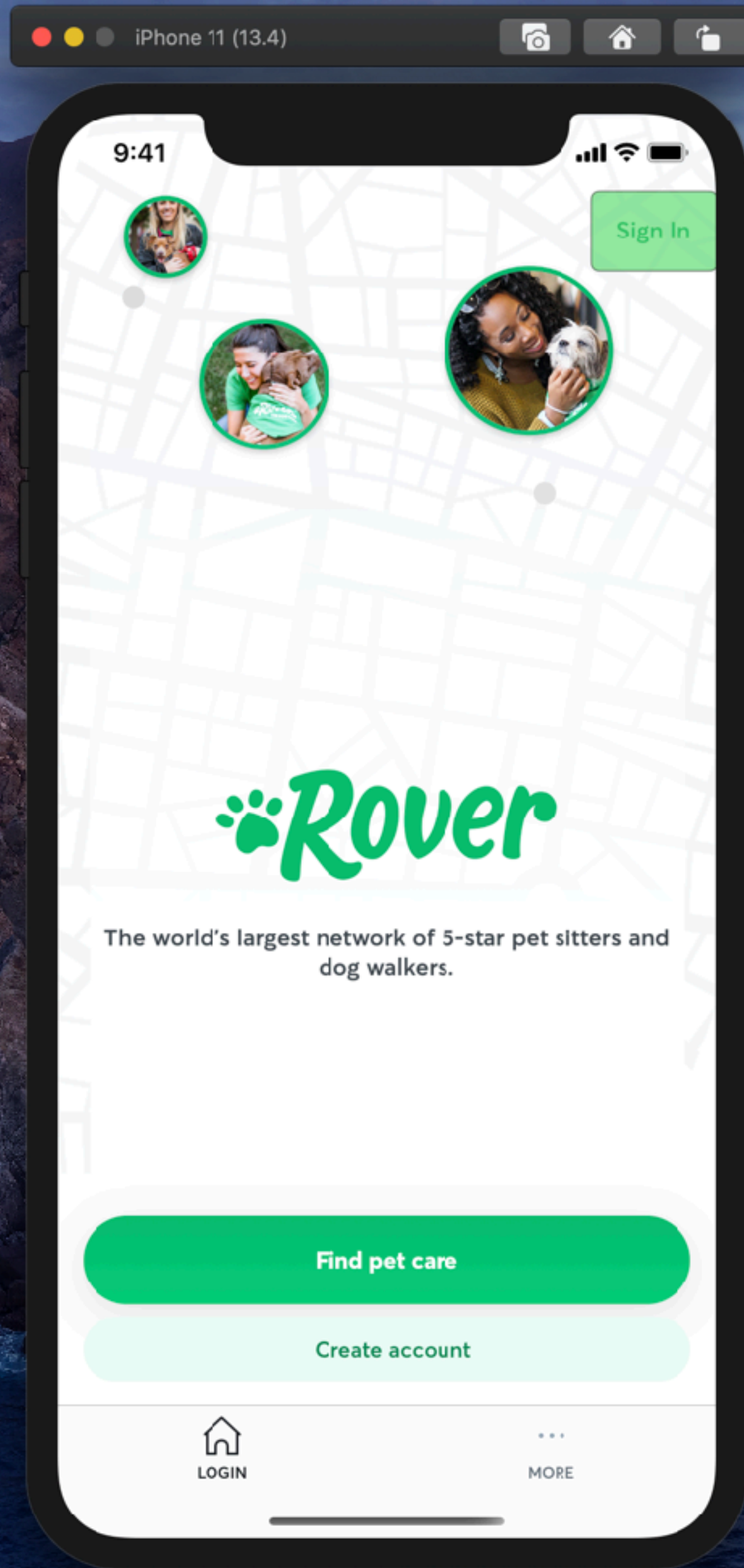
Class UIButton

Address 0x7fb4f6c174f0

Controller <nil>

Hierarchv Show

⌘Space toggles inspection pointer



Demo

Links

[**Stevie Wonder Quote**](#)

[**Accessibility Inspector \(WWDC 2019\)**](#)

[**Visual Design and Accessibility \(WWDC 2019\)**](#)

[**Dynamic Type Sizes**](#)