

Alert Controllers

AlertDialog - Learning Outcomes

Theory:

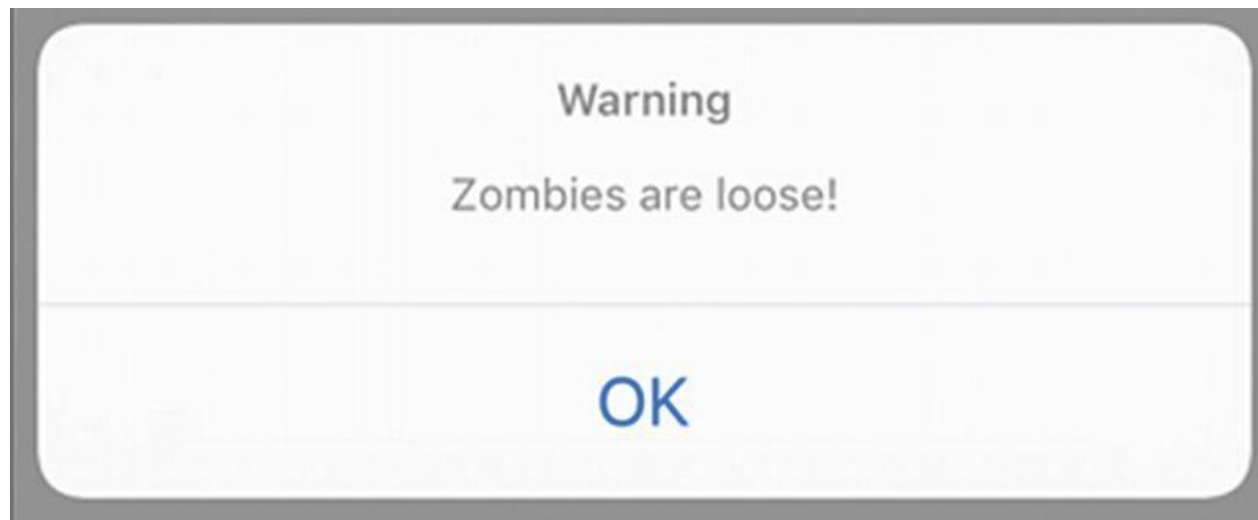
- What is an alert box
- Where are alert boxes used?

Code:

- Changing the **title**, **message**, and **buttons**
- Configuring the **style** of the alert box
- Showing the alert box to the user
- Accepting user input through the text box
- Configuring the **button handlers** for the alert box

Alert Controller

Displays information in a popup box.



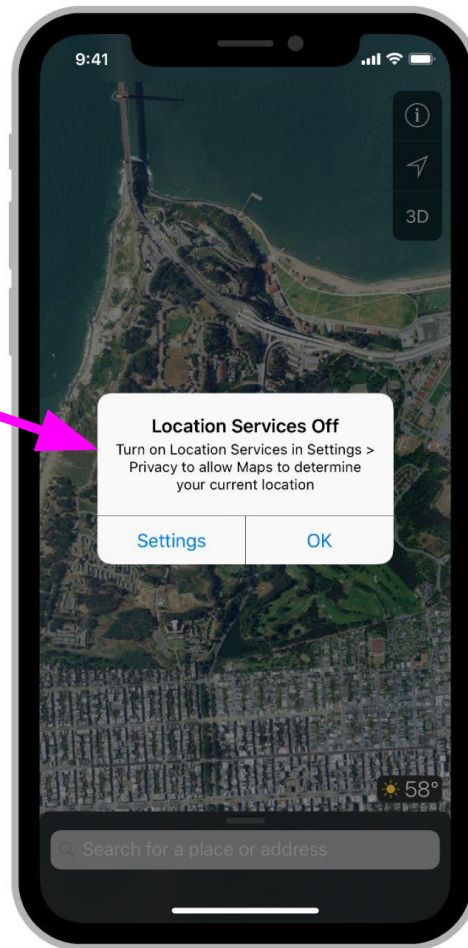
Alert Controller can be customized:

1. Customize **title**, **message**, and **number of buttons**
2. Customize **the *preferred style*** of the controller (**popup** vs. **action sheet**)

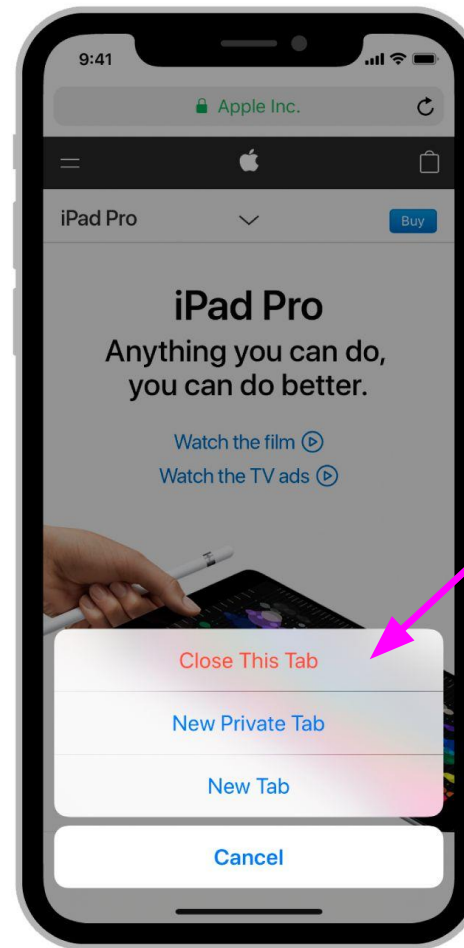


Popup vs. ActionSheet

Popup
(alert)



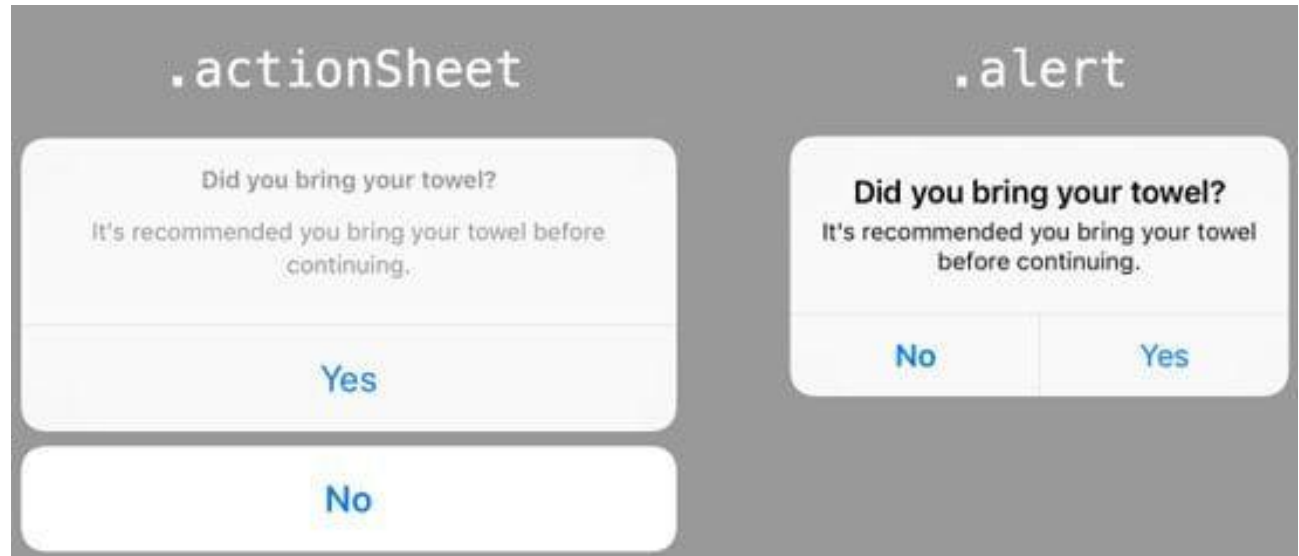
Action
Sheet



How to style your box

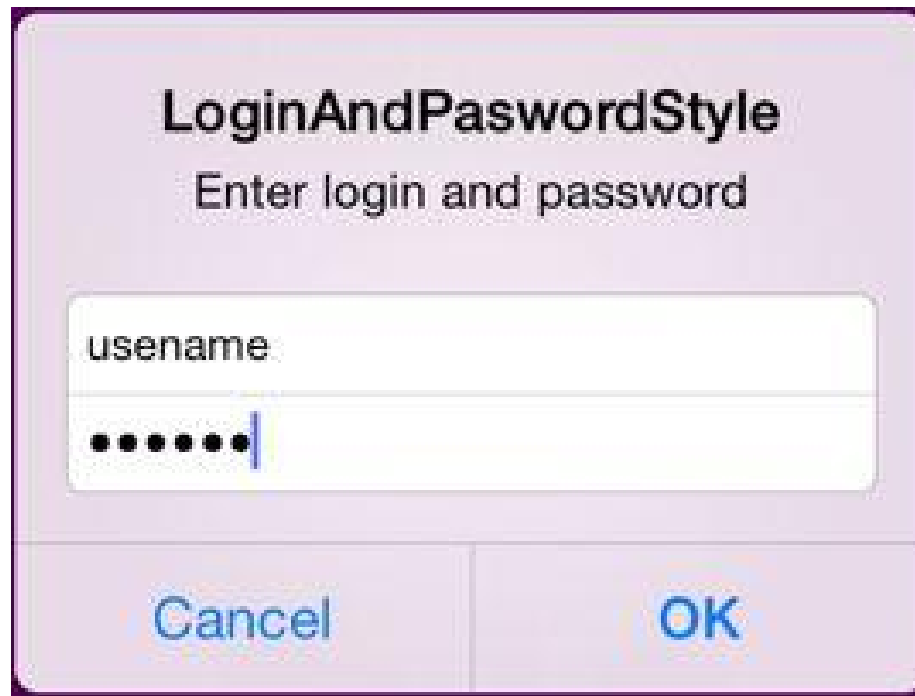
To choose the style of your popup, IOS provides two *constant* values:

- **.actionSheet**
- **.alert**



AlertControllers can accept user input

You can add a textbox to your UIAlertController:



The image shows a UIAlertController with a light purple background and a dark purple border. The title is "LoginAndPaswordStyle" in bold black text. Below the title is the subtitle "Enter login and password" in a smaller black font. There are two text input fields: the top one contains the text "username" and the bottom one contains seven black dots, indicating a password field. A blue vertical cursor is visible at the end of the password field. At the bottom of the alert, there are two buttons: "Cancel" on the left and "OK" on the right, both in blue text.

Recipe Alert - How to Create the Alert Controller

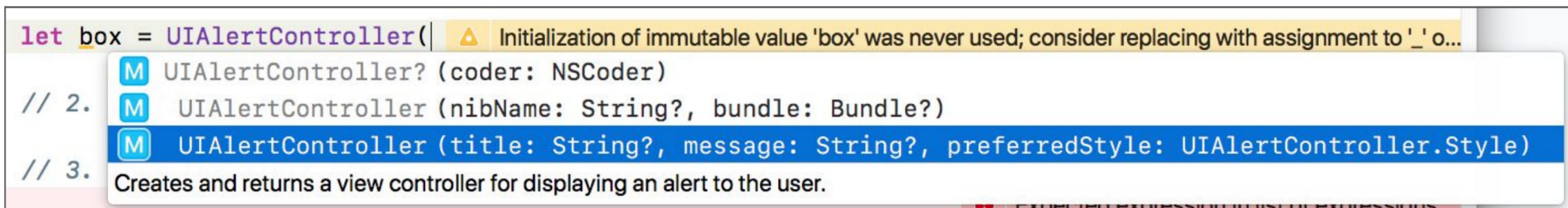
1. Create and configure the alert box
2. OPTIONAL: add buttons
3. OPTIONAL: add button handlers
4. OPTIONAL: add text fields
5. Show the alert box

1. Creating the UIAlertView

Use the UIAlertController initializer to create new Alert Box:

```
let box = UIAlertController(title: String?, message: String?,  
preferredStyle: UIAlertController.Style)
```

Use XCode's Autocomplete feature to find the correct initializer:



2. Adding an Optional Button

Buttons are optional and can be added using the `.addAction` function

```
let box = UIAlertController(title: "Weather Popup", message: "Will it rain today?", preferredStyle:
.actionSheet)
box.addAction(UIAlertAction(title: "Yes", style: .default, handler: nil))
```

The `.addAction` function accepts a *UIAlertAction*, the built-in iOS class for Alert Controller buttons

3. Adding a Button Click Handler

The UIAlertAction button can accept a call back function.

A call back function is a function that is called **after** an activity is completed

```
box.addAction(  
    UIAlertAction(title: "YES", style: .default, handler: ((UIAlertAction) -> Void)?)  
)
```

In the above snippet, the *handler* is run when the person presses “YES”

- In this case, the **callback function** is similar to a “click handler” for the YES button

4. Modifying UI from Within a Button Click Handlers

To modify UI from inside a button click handler, you must include the **self** reference

```
box.addAction(UIAlertAction(title: "Push me!", style: .default, handler: {  
    action in  
    self.resultsLabel.text = "HELLO WORLD!"  
})))
```

5. Adding a Text Box