UINavigation and UIStoryboardSegue

UINavigation

- Basic, stack based navigation

- Implemented as an array of ViewControllers

- Managed with 'push' and 'pop'

UINavigation

- Init with a "root" ViewController

```swift
func pushViewController(_ viewController: UIViewController, animated: Bool)

func popViewController(animated: Bool) -> UIViewController?

func popToViewController(_ viewController: UIViewController, animated: Bool) -> [UIViewController]?

func popToRootViewController(animated: Bool) -> [UIViewController]?
```

- Storyboard implements these as Segues

UINavigationController

- Subclass of **UIViewController**

- Is a container ViewController

- Contains a **UINavigationBar**

  - Child utilizes **UINavigationItem**

- *"Back"* Button provided for popping presented ViewController

UIStoryboardSegue

- This class supports the standard visual transitions available in **UIKit**

- Segue objects contain information about the view controllers involved in a transition

- Can be subclassed to define custom transitions between the view controllers in your storyboard file

- You do not create segue objects directly. Instead, the storyboard runtime creates them when it must perform a segue between two view controllers

UIStoryboardSegue cont'd

When a segue is triggered, the current view controller calls

```swift
func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    let currentViewController = segue.source
    let destinationViewController = segue.destination

    // This function is called in currentViewController
    // Therefore, currentViewController is already in scope
}
```

So that it can pass any needed data to the view controller that is about to be displayed before the visual transition occurs

UIStoryboardSegue cont'd

- A segue may be initiated programmatically by calling

```
performSegue(withIdentifier: "MySegueIdentifier", sender: Any?)
```

- An identifier must be supplied to the segue object. For segues that are defined in storyboard, this is done in the attribute inspector

- A sender object may be supplied, which is then available when **prepare(for:sender:)** is called

present(_:animated:completion:)

```swift
func present(_ viewControllerToPresent: UIViewController,
      animated flag: Bool,
   completion: (() -> Void)? = nil)
```

- Instance method on `UIViewController` subclasses.

- Presents a view controller modally.

- Takes in a view controller to present.

- Has animated flag.

- Provides a completion block.

  - Code executed when presentation is finished.

show(_:sender:)

```swift
func show(_ vc: UIViewController,
    sender: Any?)
```

- Instance method on `UIViewController` subclasses.

- Presents a view controller in a primary context.

  - Via navigation stack `if navigationController != nil`.

  - Modally otherwise.

- Takes in a view controller to present.

- Can optionally pass in a `sender`.

dismiss(animated:completion:)

```swift
func dismiss(animated flag: Bool,
    completion: (() -> Void)? = nil)
```

- Instance method on `UIViewController` subclasses.

- Dismisses the view controller that was presented modally.

- Has animated flag.

- Provides a completion block.

  - Code executed when presentation is finished.