SpaceAdventure
Lesson 8

Description

Add a `[Planet]` array property to the `PlanetarySystem` class to represent a collection of planets. Introduce a `Planet` class into the project.

```
Welcome to the Solar System!
There are 0 planets to explore.
What is your name?
Jane
Nice to meet you, Jane. My name is Eliza, I'm an old friend of Siri.
Let's go on an adventure!
Shall I randomly choose a planet for you to visit? (Y or N)
Y
Ok! Traveling to...
```

Learning Outcomes

- Practice declaring Swift properties and basic class definitions.
- Discover the Swift `Array` collection type and relate it to an ordered collection of objects.
- Extend the understanding of property initialization with `Array` initialization syntax.
- Practice using the Xcode Documentation and API Reference to view technical documentation.

Vocabulary

array	bracket	property
class definition	initialization	initializer
Documentation and API Reference		

Materials

- **SpaceAdventure Lesson 8** Xcode project

Opening

How can we create a property of the `PlanetarySystem` that represents a collection of planets?

Agenda

- Discuss how all planetary systems have a collection of planets, and how one might represent a collection of planets as a property of a `PlanetarySystem`.
- Discuss how an array can be used to contain an ordered collection of objects.
- Explain the concept of Swift arrays and collection types.
- Add a new property to the `PlanetarySystem` class for an array of Planet objects.

```
let planets: [Planet]
```

- Explain the components of the property declaration, emphasizing the type annotation with brackets for Swift arrays.
- Discuss the Xcode error, and how the array has a type of `[Planet]` but no `Planet` class yet exists.
- Add a new Swift file (⌘N) called **Planet.swift**, observe that the Project Navigator (⌘1) displays the new file within the SpaceAdventure group, and implement a basic `Planet` class definition.

```
class Planet {

}
```

- Using the Project Navigator (⌘1), select **PlanetarySystem.swift**.
- Discuss how the original error has disappeared, and discuss the presence of a new error.
- Discuss why the new `planets` array property, declared as a constant, must be assigned a value in the initializer.
- Update the `PlanetarySystem` initializer to expect a `[Planet]` array to initialize the `planets` property.

```
init(name: String, planets: [Planet]) {
   self.name = name
   self.planets = planets
}
```

- Discuss the named parameter syntax and the `[Planet]` array type annotation.
- Using the Project Navigator (⌘1), select **SpaceAdventure.swift**, and observe the errors in the editor.
- Discuss how the instantiation of the `PlanetarySystem` object must now match the expectations of the updated initializer.
- Update the initialization of the `planetarySystem` property.

```
let planetarySystem = PlanetarySystem(name: "Solar System",
   planets: [Planet]())
```

- Explain the array initializer syntax, and how the anonymous array is passed as an argument to the `PlanetarySystem` initializer.
- Discuss how one might use the size of a `PlanetarySystem` `planets` property to determine how many planets there are to explore in `displayIntroduction`.
- Using the Xcode Documentation and API Reference (⇧⌘0), explore the Swift Standard Library documentation for the `Array` `count` property.
- Update the implementation of `displayIntroduction` to use the `PlanetarySystem` `planets` property to determine the number of available planets to explore.

```
private func displayIntroduction() {
   println("Welcome to the \(planetarySystem.name)!")
   println("There are \(planetarySystem.planets.count) planets to
      explore.")
}
```

- Run the program (⌘R), and observe the console (⇧⌘C) output of "0 planets to explore."

Closing

Why is the program stating that there are no planets to explore? What do you think is the next feature we need to add to our program?

Modifications And Extensions

- Remove the initialization of the `planetarySystem` property, and implement a `SpaceAdventure` initializer that initializes the `planetarySystem` property.

- Declare the `planetarySystem` property as an optional `PlanetarySystem`, and update the logic in the program to handle cases where the `planetarySystem` property has no value.

Resources

The Swift Programming Language: About Swift https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/Swift_Programming_Language/

The Swift Programming Language: A Swift Tour https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/Swift_Programming_Language/GuidedTour.html

The Swift Programming Language: The Basics https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/Swift_Programming_Language/TheBasics.html

Project Navigator Help: Adding a New File https://developer.apple.com/library/ios/recipes/xcode_help-structure_navigator/articles/Adding_a_New_File.html

The Swift Programming Language: Classes and Structures https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/ClassesAndStructures.html

The Swift Programming Language: Properties https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Properties.html

The Swift Programming Language: Initialization https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Initialization.html

The Swift Programming Language: Collection Types https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/CollectionTypes.html

Swift Standard Library Reference: Array https://developer.apple.com/library/ios/documentation/General/Reference/SwiftStandardLibraryReference/Array.html