SpaceAdventure
Lesson 9

Description

Add properties and an initializer to the `Planet` class. Use a mutable array when creating the array of `planets`, and add one `Planet` to the array.

```
Welcome to the Solar System!
There are 1 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 properties within a class definition.
- Extend existing code to accommodate new features.
- Practice implementing parameterized initializers.
- Define the concepts of mutability and immutability, and relate mutability to `var` and `let`.
- Discover how to append objects to an array.

Vocabulary

property	type annotation	initializer
parameter	`self`	array
mutability	immutability	

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License, by Yong Bakos.

Materials

- **SpaceAdventure Lesson 9** Xcode project

Opening

What do we need to do to add a `Planet` to our `PlanetarySystem`? What kinds of properties does a `Planet` have?

Agenda

- Discuss the need to add properties to the `Planet` class.
- Using the Project Navigator (⌘1), select **Planet.swift**, and add two properties to the `Planet` class.

```
class Planet {

   let name: String
   let description: String

}
```

- Discuss the property declarations, the use of `let`, and the type annotations.
- Discuss the Xcode error, and the need to implement an initializer for the `Planet` class.
- Add a parameterized initializer to the `Planet` class.

```
...
   let description: String

   init(name: String, description: String) {
      self.name = name
      self.description = description
   }

}
```

- Review the concepts of initializers and initialization.
- Explain how the `Planet` initializer expects two `String` values, one called `name` and one called `description`; how the initializer assigns the value of the `name` parameter to the `name` property and the value of the `description` parameter to the `description` property; and uses `self` to refer to the object itself.

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License, by Yong Bakos.

- Using the Project Navigator (⌘1), select **SpaceAdventure.swift**.
- Discuss how a `SpaceAdventure` initializer might take responsibility for preparing the `PlanetarySystem`, by creating `Planet` objects and adding them to the `planetarySystem` property's `planets` array.

```
init() {
   let mercury = Planet(name: "Mercury", description: "A very hot
      planet, closest to the sun.")
   planetarySystem.planets.append(mercury)
}
```

- Discuss the Xcode error, and why a `Planet` cannot be appended to the `planets` array because the array is immutable.
- Using the Project Navigator (⌘1), select **PlanetarySystem.swift**.
- Modify the `planets` property declaration, replacing `let` with `var`.

```
var planets: [Planet]
```

- Explain that Swift supports both mutable and immutable arrays, and how `var` and `let` indicate mutability and immutability.
- Run the program (⌘R), and observe the console (⇧⌘C) output displaying that there is "1 planet to explore."

Closing

Can you figure out how to add the other eight planets of our solar system to the `planets` array?

Modifications And Extensions

- Modify the `SpaceAdventure` initializer such that the `Planet` object is created and passed to `append`, without assigning it to a constant first.
- Enhance the `Planet` model such that a `Planet` can have multiple moons.
- Enhance the `displayIntroduction` method to correctly pluralize the word "planet" based on the value of `planetarySystem.planets.count`.

Resources

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

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License, by Yong Bakos.

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

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License, by Yong Bakos.