# SpaceAdventure
## Lesson 6

## Description

Extract the work done by the `start` method into three well-named methods.

```
Welcome to our solar system!
There are 8 planets to explore.
You are currently on Earth, which has a circumference of 24859.82 miles.
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)
Huh?
Sorry, I didn't get that.
Shall I randomly choose a planet for you to visit? (Y or N)
Y
Ok! Traveling to...
```

## Learning Outcomes

- Reorganize related or repetitive code into well-named methods, and analyze the resulting expressiveness and readability of code.
- Discover how methods can call other methods defined within the same class definition.
- Discover the `return` keyword and apply it in a method that returns a value to its caller.
- Recognize how Swift method parameters are declared and passed.

## Vocabulary

| | | |
|---|---|---|
| method | method call | access control |
| `private` | parameter | type annotation |
| argument | return type | `return` |

# Materials

- **SpaceAdventure Lesson 6** Xcode project

# Opening

What general tasks does the `start` method accomplish? What repetitive code do we see?

# Agenda

- Discuss how the `start` method seems to do three things: print an introduction, greet the user, and determine which planet to travel to.
- Extract the first few lines of `start` into a new private method called `displayIntroduction`.

```
private func displayIntroduction() {
   let numberOfPlanets = 8
   let diameterOfEarth = 24859.82 // In miles, from pole to pole.
   println("Welcome to our solar system!")
   println("There are \(numberOfPlanets) planets to explore.")
   println("Your are currently on Earth, which has a circumference
      of \(diameterOfEarth) miles.")
}
```

- Replace the extracted code with a method call at the beginning of `start`.

```
displayIntroduction()
```

- Discuss how the `displayIntroduction` will only be called by the `start` method, and is marked `private` to indicate that only code within the same file will be able to call `displayIntroduction`.
- Briefly explain the three Swift access levels: public, internal and private.
- Discuss how the `start` method uses a pair of `println` and `getln` methods twice, to prompt for and capture user input.
- Encapsulate the work of prompting for and capturing user input into a private method called `responseToPrompt`.

```
private func responseToPrompt(prompt: String) -> String {
   println(prompt)
   return getln()
}
```

- Explain the method implementation syntax, emphasizing the parameter name, parameter type annotation, and the method return type.
- Discuss how this method will print the `String` it receives, and will return what the user types as a `String`.
- Replace the relevant lines of code in `start` to use the new `responseToPrompt` method.

```
let name = responseToPrompt("What is your name?")
println("Nice to meet you, \(name). My name is Eliza, I'm an old
   friend of Siri.")

println("Let's go on an adventure!")

var decision = "" // Start with empty String
while !(decision == "Y" || decision == "N") {
   decision = responseToPrompt("Shall I randomly choose a planet for
      you to visit? (Y or N)")
   if decision == "Y" {
...
```

- Discuss the function (or method) calling syntax with the use of an argument, tracing how the string literal argument passed to `responseToPrompt` becomes assigned to the method's `prompt` parameter, and is used by the body of the method.
- Extract the greeting-related code in `start` into a new method called `greetAdventurer`.

```
private func greetAdventurer {
   let name = responseToPrompt("What is your name?")
   println("Nice to meet you, \(name). My name is Eliza, I'm an old
      friend of Siri.")
}
```

- Extract the code remaining code in `start` into a new method called `determineDestination`.

```
private func determineDestination() {
   var decision = "" // Start with empty String
   while !(decision == "Y" || decision == "N") {
      decision = responseToPrompt("Shall I randomly choose a planet
         for you to visit? (Y or N)")
      if decision == "Y" {
         println("Ok! Traveling to...")
         // TODO: travel to random planet
      } else if decision == "N" {
         println("Ok, name the planet you would like to visit...")
         // TODO: let the user select a planet to visit
      } else {
         println("Sorry, I didn't get that.")
      }
   }
}
```

- Update the `start` method to call the new `greetAdventurer` and `determineDestination` methods.

```
func start {
   displayIntroduction()
   greetAdventurer()
   println("Let's go on an adventure!")
   determineDestination()
}
```

- Run the program (⌘R), observe the console(⇧⌘C), and confirm that the functionality remains unchanged.
- Discuss whether or not the `start` method has become more readable, expressive and concise.

# Closing

Take a look at our `TODO` comments in `determineDestination`. What nouns do you see? What else do you think our program needs to model?

# Modifications And Extensions

- Investigate the difference between functions and methods.
- Add the `private` access control modifier to the `start` method definition, re-run the program, and explain the reason for the error you see.

# 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

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

The Swift Programming Language: Access Control https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/Swift_Programming_Language/AccessControl.html

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