

SpaceAdventure

Lesson 3

Description

Add a `while` loop to validate the user input of Y or N.

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.

Y

Ok! Traveling to...

Learning Outcomes

- Relate repetitive tasks to Boolean logic and control flow with a `while` loop.
- Practice applying Boolean logic with an expression that uses `not` and `or`.

Vocabulary

input validation	Boolean/logical operator	<code>not</code>
<code>or</code>	loop	<code>while</code>
<code>else if</code>		

Materials

- SpaceAdventure Lesson 3 Xcode project

Opening

How can we properly handle Y, N and everything else the user might type?

Agenda

- Discuss the need to ask the traveler if he wants to visit a random planet, and to prompt for another answer "as long as the traveler does not answer Y or N."
- Modify the existing user input capturing and decision making to leverage a `while` loop to carry out the repetitive task of waiting for the user to type Y or N.

```
var decision = "" // Start with empty String
while !(decision == "Y" || decision == "N") {
    decision = getln()
    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.")
    }
}
```

- Explain `while` loop syntax, the `!` operator for logical negation, parenthesis for grouping expressions, and the `||` operator.
- Discuss why `decision` is now a variable, as it may be assigned different values during repetitions of the loop.
- Discuss the `if` statement, the `else if` clause and the `else` clause.
- Run the program (⌘R), interact with the console (⇧⌘C), try some arbitrary input, and observe that the program continues to prompt until Y or N is entered.

Closing

Read the code aloud as if it were a story. It seems to be about `println`, `getln` and loops. How can we introduce the concept of an "adventure" into our program?

Modifications And Extensions

- Add an "easter egg" to the program, so that when the user types something like `Cookies!` instead of `Y` or `N`, the program responds accordingly.

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: While Loops https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/ControlFlow.html#//apple_ref/doc/uid/TP40014097-CH9-ID124

The Swift Programming Language: Logical Operators https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/BasicOperators.html#//apple_ref/doc/uid/TP40014097-CH6-ID76

The Swift Programming Language: Conditional Statements https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/ControlFlow.html#//apple_ref/doc/uid/TP40014097-CH9-ID127