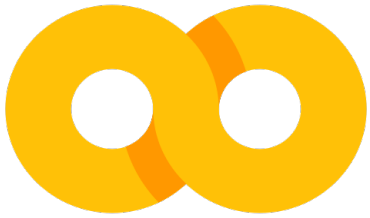
How to Make an App For Beginners

Module 2
Lesson 7
Worksheet



Welcome!

In this worksheet, you'll practice working with Loops!

Just in case you don't remember how you got here (or if you need a refresher), Lesson 7 can be found here:

https://youtu.be/3B8joVTWVtc

Step 1: We'll be doing these exercises in a Swift Playground.

Open Xcode and create a new playground
(File Menu->New->Playground).
From the list of Playground templates, just select "**Blank**"

Step #2: Let's declare a basic array and dictionary which we will use for this worksheet. Type out the following:

```
// Dictionary declaration
var dict = ["CA":"California", "TX":"Texas"]

// Array declaration
var arr = ["AL", "UT", "NY"]
```

Step #3: Let's practice some basic For-In loops.

```
// Basics For In Loops

// no counter variable
for _ in 1...2 {
    print("hello")
}

// with counter variable
for someVar in 1...10 {
    print(someVar)
}
```

Step #4: Let's use the for-in loop to display the contents of the array.

```
// Iterate through the array
for stateAbbrev in arr {
    print(stateAbbrev)
}
```

Now let's do it this way (Notice that the range is 0 to the count of the array minus 1):

```
for arrIndex in 0...arr.count-1 {
    print("#\(arrIndex) is \(arr[arrIndex])")
}
```

Inside the print statement above, we're simply doing variable substitution in the String which is the same thing we did in the War app when we were generating random cards.

Step #5: Let's iterate through the dictionary now.

```
// Iterate through the Dictionary
for (abbrev, state) in dict {
    print("\(abbrev) is \(state)")
}
```

In this case, we can simply print out the abbrev and state from the tuple since they're Strings, however for other data types, we might need to cast the abbrev and state variables to the appropriate data type before using it.

Casting means just telling Xcode to treat that variable as the data type you cast it to. Since sometimes Xcode has no idea what data types the key/values are in your Dictionary, you might need to specifically say that "abbrev is an Int" or "abbrev is Bool" for example.

If you're unsure, just try to use it without casting it (just like we did in this example). Xcode will give you an error/warning if they need to be specifically cast to a data type before using.

Side note: there's a data type called "Any" which means that it can be ANY data type. Pretty much a wild card. When the data type of the values in your Dictionary is "Any", Xcode will most likely ask you to cast it to a specific data type before trying to use it. The "Any" data type is useful if you want to store a variety of different data types for the values of your dictionary.

Step #6: Let's switch gears now and try out some of the other types of loops. Let's try a While loop now

```
// While loop
while arr.count <= 0 {
    arr.append("OH")
}
print(arr)
```

This while loop will add "OH" into the array so long as its count is equal to or less than 0.

Notice that this While loop doesn't do anything because the array's count is 3 already prior to reaching this While loop.

Step #7: Let's try a Repeat While loop now

```
// Repeat while
repeat {
    arr.append("OH")

} while arr.count <= 0
print(arr)
```

This Repeat While loop uses the same condition as the While loop you wrote in Step 6, however if you notice in the output of the print statement below it, "OH" is added into the array.

This is because the Repeat While loop is guaranteed to loop at least once since the **Repeat While loop checks the condition AFTER each loop.**

The While loop checks the condition BEFORE each loop.

You're done! If you completed this exercise, celebrate and let me know on Twitter!
https://twitter.com/CodeWithChris

Download the Playground for this worksheet here:
https://codewithchris.com/code/Module02Lesson07PlaygroundYouTube.zip