

# SWIFT DATA STRUCTURES: DICTIONARIES

Week 2 - Day 2 - Lesson 2

BY: WELLINGTON MORENO

---

## CONNECTION TO LONG-TERM OBJECTIVE

Dictionaries are a core programming competency that enable more sophisticated code, and therefore applications.

## LEARNING OBJECTIVES

- + **Articulate** the purpose of a Dictionary
- + **Create** Dictionaries with different data types
- + **Add** data to a Dictionary
- + **Iterates** keys and values in a Dictionary
- + **Identify** the role the key plays in a dictionary

## INSTRUCTOR PREP

- + Take a look at the [lab](#)
- + Open the [starter code](#)

## STUDENT PRE-WORK

Students should be able to:

- + Write functions
- + Write loops
- + Create Playgrounds in XCode
- + Declare and use variables and constants

# LESSON GUIDE

<b>TIMING</b>	<b>TYPE</b>	<b>TOPIC</b>
10	Opening	Solve this problem
10	Introduction	This is a Dictionary and why you need it
10	Demo	This is how you make dictionaries
10	Independent Practice	Create some dictionaries
10	Demo	This is how you use a dictionary
20	Independent Practice	Solve the original problem
5	Conclusion	Key takeaways
75		

# OPENING (10 mins)

Have the students start by solving a problem:

## PRACTICE PROBLEM

Write a function that takes a list of names and returns the most popular one.

---

# INTRODUCTION: DICTIONARIES (10 mins)

## DEFINITION

"Dictionaries are unordered collections of key-value associations."

- Swift Reference

## ANALOGIES

- + **Dictionary:** How does a real work dictionary work? You look up a word, and then you find it's definition.
- + **Post Office:** You have your address, and mail gets put your unique address. No one else can have your address, so no one else can get your mail.
- + **DNS:** You put in a Human-Readable address, the system gives you back a machine address.

## KEY POINTS

- + You will notice that in a dictionary, the **key** is unique. You'll never see a word appear twice in the dictionary. A word may have multiple definitions, but it will never have more than one entry.
  - + Anything that you use as the key must be **immutable**.
  - + You can write them two different ways.
  - + You cannot add values to a constant Dictionary
  - + Dictionaries are also known as Hashmaps or Maps. Learn both names.
-

## CHECK FOR UNDERSTANDING: WRITE-SHARE

What would you use as the key if you were creating a dictionary of all of the students in the class?

What about of a water bottle? What would you use?

## DEMONSTRATION

- + Give a couple of visual models of what a dictionary is like.
  - + Show the syntax for creating a dictionary
  - + Show the syntax for reading a dictionary
- 

## DEMO: CREATING DICTIONARIES (5 mins)

Create a dictionary mapping a phone to its price.

## INDEPENDENT PRACTICE (10 mins)

Create a dictionary with at least 5 values that maps:

1. A word to its definition
2. A book to its author
3. A person to their age
4. A zip code to its state
5. An album to its tracks
6. A sports team to their players

---

# DEMO: USING DICTIONARIES (5 mins)

For this example, use a Dictionary of phone prices.

## ACCESSING VALUES

Show how to get values using keys.

## ADDING VALUES

Create a student fear dictionary that maps a student to their greatest fear.  
Show code adding some additional names

## REMOVING VALUES

Show code that removes a few entries from the Dictionary.

## ITERATION

Show code where I can access and print the values in a dictionary.

# INDEPENDENT PRACTICE (20 mins)

## SOLVE THESE PROBLEMS

- + Write a function that takes a Dictionary: [Double, Int] and returns a copy of it.
  - + Write a function that takes an array of words and returns the word that appeared the most
- 

# CONCLUSION (10 mins)

## TAKEAWAYS

- + Dictionaries allow you to form relationships in your data.
- + A Database is a type of complicated Dictionary, that allows you to find a value for a query ( ).

## BEST PRACTICES

- + Always check if the dictionary has the key you want before doing setting or retrieving it
- + Name them for the type of data they contain
- + Use a unique key