GENERAL ASSEMBLY

MOBILE DEVELOPMENT

LESSON 12
SWITCH, ENUMS, PERSISTENCE INTRO

Tedi Konda
VP, Engineering and Technology, RepEquity

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- Switch statements
- Enumerated types
- Intro to persistance
- Work on mid class projects

SWITCH STATEMENT

SWITCH STATEMENT

- switch statements are like if-else statements, but usually depend on one primary condition (known as a **control expression**) that is evaluated over a large range of possibilities.

- Other notable keywords:
  - case
  - break
  - default
  - where

CASE

- case
  - Defines a pattern/result for the control expression. If it’s true, the code after case is evaluated.

DEFAULT

- default
  - Code after this keyword is hit if all other patterns/results defined by the case keyword are not satisfied.

WHERE

- where
  - Code after this keyword is evaluated if all other patterns/results defined by the case keyword are not satisfied.

ENUMERATIONS

ENUM

- An enumeration, or enum, is a way to group related values together.

- To Playgrounds!

GETTING STARTED

INTRODUCTION TO PERSISTENCE

WHAT IS PERSISTENCE AND WHY DO IT?

- We persist data so that we can access it quickly between sessions.
- Examples
  - App high scores
  - App settings
  - User credentials
  - Etc.

HOW DO YOU PERSIST? (PT. 1)

- There are many ways to persist data
- Choosing how to persist depends on several things
  - What kind of data am I writing?
  - What kind of data am I reading?
  - Am I storing relations between things?
  - For how long do I need store the data?

HOW DO YOU PERSIST? (PT. 2)

- There are several built-in options for persisting data in iOS
  - This session:
    - User defaults
    - Property lists
    - Flat files
  - Next session:
    - Core Data
    - SQLite

USER DEFAULTS

USER DEFAULTS

- A key/value store for storing small, independent bits of data

- What else have we used that utilizes the the key/value paradigm?

USER DEFAULTS

- What kind of data am I writing?
  - Small bits of data and an associated key, stored one at a time
  - e.g. A string, a number, a boolean, a dictionary, an array, a date, etc.
- What kind of data am I reading?
  - Same as above, retrieved one at a time
- Am I storing relations between things?
  - No
- How persistent does my data need to be?
  - Persistent across app sessions, but is deleted when the app is deleted

USER DEFAULTS

- Good for:
  - App settings
  - App state
- Not good for:
  - Large data sets
  - Complex relations
  - Sensitive data
  - Caches

NSUSERDEFAULTS

- To Playgrounds!

NSUSERDEFAULTS (ADVANCED)

- If you want to store a custom class, you'll need to use the NSKeyedArchiver and NSKeyedUnarchiver class.
- The NSKeyedArchiver class allows you to convert your class (and its properties) into NSData. You then save NSData into your project.
- The NSKeyedUnarchiver class allows you to build your custom class with values saved as NSData inside of NSUserDefaults.

- More info in NSUserDefaults section of http://nshipster.com/nscoding/