Class 04 (06/15/2015)

Operators, Optionals, and Functions

Lesson 3 Review

* Nomenclature
* Syntax

Inclass Assignment Review

* If -> if the condition is true, it will stop the execution
* && -> and
* || -> or
* what inside the forloop -> not executable outside of the forloop
* arc4random
* if – else statements: set the most demanding conditions first
* Fibonna

Operators

* Elements = variables or constants
* Unary Operators -> acting on one element
  + == x
  + –x
  + x = true; !x //false
* Binary Operators -> actin on two elements
  + Arithmetic operators + = \* /
  + comparison operators
  + Logical AND/OR operators
    - &&
    - ||
* Ternary Operators
  + Ternary Conditional Operators
    - (x > 0) ? stringTrue : stringFalse
    - let z = (x > 0) ? stringTrue : stringFalse

Optionals

* Typically, constants and variable have vlues
* But there may be a situation where you may not yet know the value of your constants or variables
* Set it to be empty – nil
  + Empty variables
* Example 1)
  + let x = 10
  + var y: Int? //type? = nil
  + y = 10 // now y has a value of 10
* Example 2)
  + var age: Int? // age = nil

Optional Binding

* action of checking whether something is nil
  + ex) check whether it has an value or a nil
    - if (let unwrappedAge : Int) = age {
    - “The Person is \(unwrappedAge) years old”
    - } else {
      * “The variable, ‘ age’ , was not set.”
    - }
    - // unwrappedAge is only valid within the scope of the block
      * covered in for-in-loop
    - so in fact, you can use age instead of unwrappedAge -> Swift has two ‘age’s but Swift overrides the former ‘age’

Forced Unwrapping (“Dead as certain it has some value or is not nil”)

* ex1)
  + var name: String? //nil
  + name! //nil
* ex2)
  + var name: String? = Arthur
  + name! //will crash
* important because your program has 1,000 + lines
* ex3)
  + var age: Int? = 10
  + if let \_ = age {
    - println(“age: \(age)”)
  + } else {
    - age = 100
  + }
* Arthur’s Medium Article

Function

* Fifth test function
  + //new variable: statement
  + // why returning string? (mistake) but useful when you search username in database -> if not found -> nil -> your username isnot
* “deferred initialization” = look for it
* why use functions?
* Common conventions
  + Descriptive function names
  + Keep the content of your functions under 50 lines (if possible)
  + Make your functions abstract
  + Two principles to keep in mind
    - KISS: Keep IT Simple Stupid
    - DRY: Don’t Repeat Yourself
      * You need to type your name in one function and your name and age in another function (nil, optional to create one function)