Swift for beginners:

1. **Variables and Types:**
   1. Intro to Variables

Biến là một giá trị dữ liệu có thể được thay đổi trong chương trình. Giá trị dữ liệu có thể là loại chuỗi (string), số tự nhiên (interger), số trôi nổi (float), logic (boolean), mảng (array), đối tượng (object) hoặc dữ liệu rỗng (NULL).

* 1. Data types

**Kiểu dữ liệu**là một thuật ngữ chỉ mỗi loại dữ liệu sẽ có một cách hoạt động riêng do hệ thống định nghĩa.Ví dụ :Nếu muốn tính toán + – \* / thì bạn phải dùng kiểu số (Int,Float,…) chứ không thể dùng kiểu chuỗi(String),….Ngoại trừ trường hợp bạn sử dụng thuật toán để thực hiện tính toán + – \* / hai chuỗi với nhau.Tuy nhiên trong thuật toán cũng phải dùng kiểu Int ,… để xử lý tính toán.

|  |  |  |
| --- | --- | --- |
| **Swift** | **Java** | **x86 Assembly** |
| var myNumber: Int | int myNumber; | myNumber DD ? ; myNumber is an Int\* |

Các kiểu dữ liệu cơ bản trong Swift:

|  |  |  |
| --- | --- | --- |
| **Type** | **Description** | **Example Values** |
| Int | whole number values | 0, 2, -2, 100, ... |
| Float | floating-point (decimal) numbers | 3.14, 5.693, -12.321, ... |
| Double | floating-point (decimal) numbers that have even more precision—use this instead of Float | 3.14, 5.693, -12.321, ... |
| Bool | a truth value | true, false |
| Character | a single letter, digit, symbol, or code | "a", "+", "q", ... |
| String | combinations of characters | "swift", "I Love Swift", " ", ":)", ... |

* 1. Creating Variables

Một vài ví dụ khai báo:

**var** petsAge = 12

**var** myMiddleInitial: **Character** = "A"

**var** numberOfWheels: **Int** = 4

**var** centimetersOfRainfall: **Float** = 5.5

**var** pi: **Double** = 3.14159265359

**var** letterOfTheDay: **Character** = "z"

**var** myFavoriteAnimal: **String** = "nudibranch"

**var** rainingOutside: **Bool** = true

* 1. Changing a Variable’s Value

Để thay đổi giá trị của biến, chúng ta sử dụng cú pháp sau đây:

**{nameOfVariable} = {newValue}**

Ví dụ ta thay đổi giá trị của biến **petsAge** từ 13 thành 14:

**petsAge = 13**

và

**petsAge = 14**

* 1. Constants

Một điểm chú ý là giá trị constant không thể bị thay đổi. Nó được khai báo như sau:

Ví dụ:

**let** encouragement = "You can do it!"

Điều gì sẽ xảy ra với đoạn code bên dưới

**let** encouragement = "You can do it!"

encouragement = "C'mon, you got this!"

**Nó sẽ xuất hiện lỗi như sau**

***compiler****: ERROR: Cannot assign a value: 'encouragement' is a 'constant'*

* 1. Naming Variables and Constants

1. **String:**
   1. Intro to strings
   2. Xcode Playgrounds
   3. String Initialization
   4. What’s a String Made Of?
   5. Intro to String Interpolation
   6. String Manipulation
   7. String Recap
2. **If statement:**
   1. Boolean Expressions
   2. && and ||
   3. The ! Operation
   4. Intro to if Statements
   5. If Statements Pratice
   6. If Else and Else If statements
3. **Functions:**
   1. Function Syntax
   2. Function with Parameters
   3. Function with Multiple Parameters
   4. Function Scope
   5. Return Values
   6. Default Parameters and Using Functions
4. **Operators And Expressions**
   1. Introduction to Operators and Expression
   2. Operators in Playgrouds
   3. Term to remember
5. **Boolean Expressions**
   1. Introduction to Boolean Expressions
   2. The == Operator
   3. The && Operator
   4. The || Operator
   5. The ! Operator
6. **Branching with If Statements**
   1. Introduction to Branching
   2. If Statements
   3. If Else & Else If Statements
   4. Nested If Statements
7. **Switch Statements**
   1. Introduction to Switch Statements
   2. Switch Syntax
   3. Fallthrough
8. **Function I**
   1. Introduction to Function I
   2. Function Syntax
   3. Functions with Parameters
   4. Function Scope
   5. Returning Values
   6. Default Parameters and Using Functions
9. **Function II**
   1. Introduction to Functions II
   2. Tuples: Multi Return Values
   3. In-Out Parameters
   4. Nested Functions
   5. Internal and External Parameter Name
   6. Designing Functions
10. **Strings**
    1. Introduction to Strings
    2. All about Strings
    3. String Interpolation
    4. String Manipulation
    5. Strings, Tying It All Together
11. **Optionals**
    1. Introduction to Optionals
    2. About Nil Values
    3. The Optional Type
    4. Unwrapping Optionals
    5. Optional Chaining
    6. Passing Optionals as Parameters to Functions
12. **Loops**
    1. Introduction to Loops
    2. For-Each Loops
    3. While Loops
    4. The break Keyword
13. **Arrays**
    1. Introduction to Arrays
    2. All About Arrays
    3. Looping Through Arrays
14. **Dictionaries**
    1. Introduction to Dictionaries

Dictonary khá giống với Array, tuy nhiên thay vì index, Dictionary quản lý các phần tử của mình bằng các cặp key-value, trong đó key là duy nhất.

* 1. Dictionary Initialization

Có 2 cách để khởi tạo Dictionary. Lấy ví dụ đơn giản là khởi tạo một Dictionary lưu tên và số điện thoại. Bạn có thể sử dụng một list các cặp key-value tách biệt nhau như sau:

var phoneNumbers = ["Jenny": "867-5309", "Mateo" : "510-7750", "Mike": "330-8004"]

Hoặc cũng có thể khởi tạo như sau:

var phoneNumbers = [String: String]()

Chú ý rằng, mối quân hệ liên kết giữa key và value được thể hiện bởi dấu hai chấm ( : ) và việc tách biệt các phần tử được thể hiện bởi dấu phẩy ( , ).

Value có thể được gán giá trị là bất cữ kiểu dữ liệu nào. Key thì lại không như thế, key chỉ có thể được khai báo là các kiểu dữ liệu nguyên thuỷ (String, Int, Double and Bool).

Chúng ta có thể truy xuất giá trị (value) của một key như sau:

phoneNumbers["Jenny"]

* 1. Dictionary Operations

Chúng ta sẽ gán thêm (insert) phần tử mới cho Dictionary bằng câu lệnh sau:

phoneNumbers["Valeria"] = "894-124"

Thay đổi giá trị của một phần tử:

phoneNumbers["Jenny"] = "867-1111"

Xoá một phần tử:

phoneNumbers.removeValue(forKey: "Jenny")

1. **Advance Operator**
   1. Getting Bit-Wise

Swift có một loại toán tử đặc biệt được gọi là “Bitwise”. Nó có tên gọi như thế là bởi vì toán tử này dựa trên kiểu bits. Bit có giá trị 0 hoặc 1. Toán tử Bitwise cho phép bạn thao tác các dữ liệu bit thô cá nhân bên trong một cấu trúc dữ liệu. Chúng thường được sử dụng trong lập trình cấp thấp, chẳng hạn như lập trình đồ hoạ và tạo điều khiển thiết bị. Hoặc khi bạn làm việc với dữ liệu thô từ mã nguồn bên ngoài, chẳng hạn như mã hoá và giải mã dữ liệu thông qua một giao thức tuỳ chỉnh.

* 1. Using Bitwise Operators
  2. A Bit Further
  3. Overloading Operators
  4. Nil Coalescing & Custom Operators

1. **Enums**
   1. Enums.

Enum được sử dụng để định nghĩa một tập hợp có số phần tử cố định và liệt kê sẵn, bạn không thể thêm hoặc bớt số phần tử.

Ví dụ khai báo một enum được sử dụng để liệt kê một số màu cơ bản:

enum PrimaryColor

{

case red

case blue

case yellow

}

* 1. Example Enums, Switch Statements and Initialization

Bạn có thể đặt ra bài toán như sau: “Hãy xem xét giá trị của PrimaryColor. Trong trường hợp nó bằng .red, thì in ra màn hình “Red Color”, tương tự với .blue và .yellow “.

Như được mô tả trong  Control Flow, một lệnh switch có thể được sử dụng để xét một phần tử Enum. Ta sẽ thực hiện bài toán trên như sau:

switch PrimaryColor

{

case .red:

println("Red Color.")

case .blue:

println("Blue Color.")

case .yellow:

println("Yellow Color.")

}

1. **Plist and JSON**
   1. Saving Data

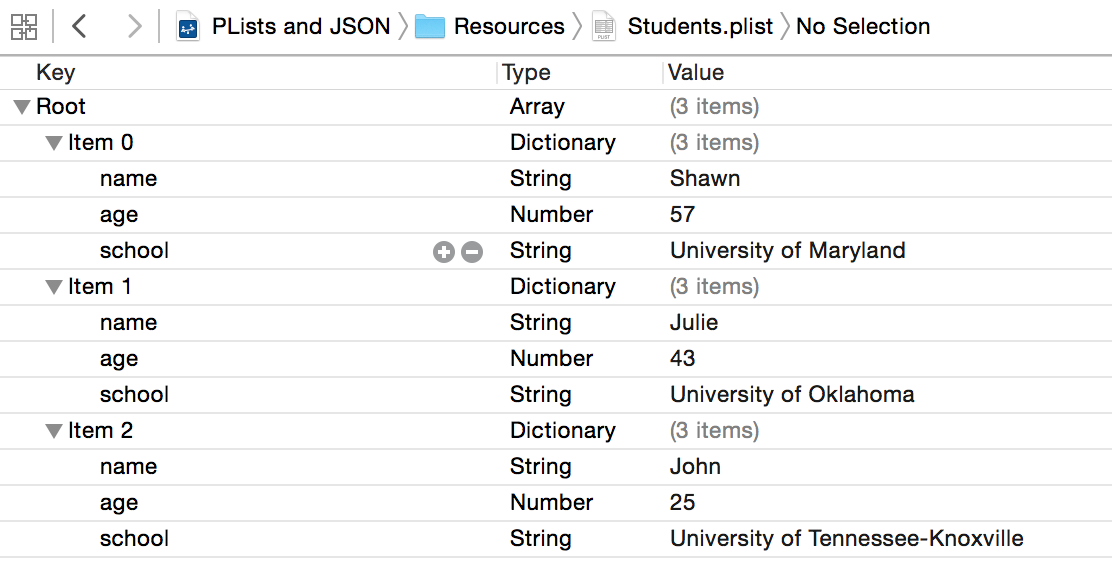
Dữ liệu luôn là thứ bắt buộc phải có của tất cả các ứng dụng. Dữ liệu được hiển thị dưới dạng các biến (variable), cấu trúc (struct) hay chuỗi (array). Tuy nhiên, những dữ liệu đó chỉ tồn tại khi code của bạn đang chạy, còn khi bạn thoát khỏi ứng dụng, toàn bộ dữ liệu đó sẽ biến mất.

Vậy, nếu chúng ta muốn dữ lại tất cả những dữ liệu đó ngay cả khi thoát khỏi ứng dụng thì phải làm thế nào? Chúng ta có một khái niệm gọi là “Persisting Data”, nó là một thuật ngữ dùng để nói về việc lưu trữ dữ liệu. Có rất nhiều cách để lưu trữ dữ liệu, và 1 trong những cách đó là lưu file trên thiết bị người dùng, theo thứ tự các bước như sau:

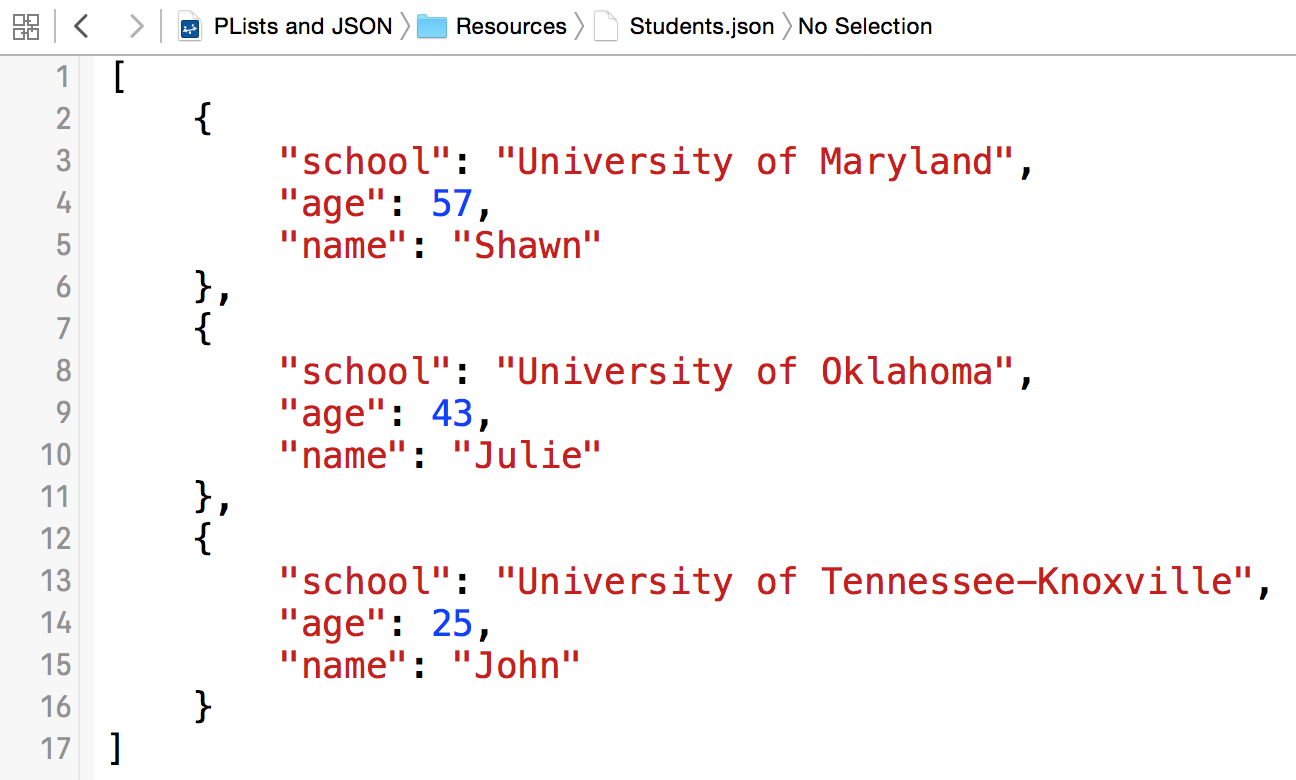
1. Kiểm tra file đã tồn tại hay chưa (ví dụ record.txt)
2. Nếu file đã tồn tại, mở file và cập nhật dữ liệu.
3. Nếu file chưa tồn tại, tạo file.
4. Đóng file lại.
5. Lấy dữ liệu của file và hiển thị dữ liệu cần thiết lên màn hình.
   1. Working with Data

Chúng ta sẽ tập trung vào 2 loại dữ liệu mà sẽ được sử dụng thường xuyên khi xây dựng ứng dụng: Plist và JSON.

Ví dụ file Plist:

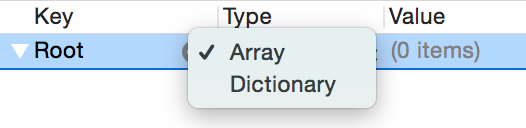


Ví dụ file JSON:

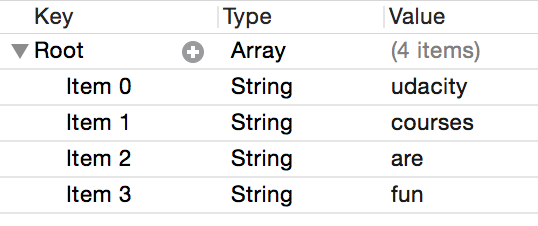


* 1. Loading a Plist

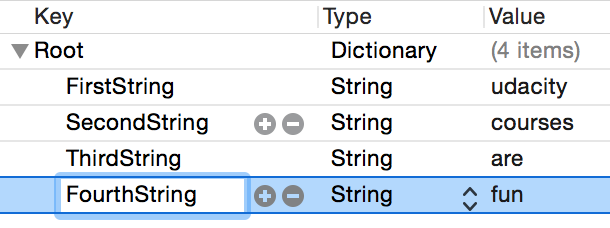
File Plist được bắt đầu bằng việc định nghĩa trong Root contrainer, nó có thể là một Array hoặc một Dictonary. Việc này rất quan trọng vì nó sẽ xác định cách chúng ta tương tác với dữ liệu này với Swift.



Nếu Root container là một Array, thì mỗi phần tử của nó được đánh số bắt đầu từ 0:



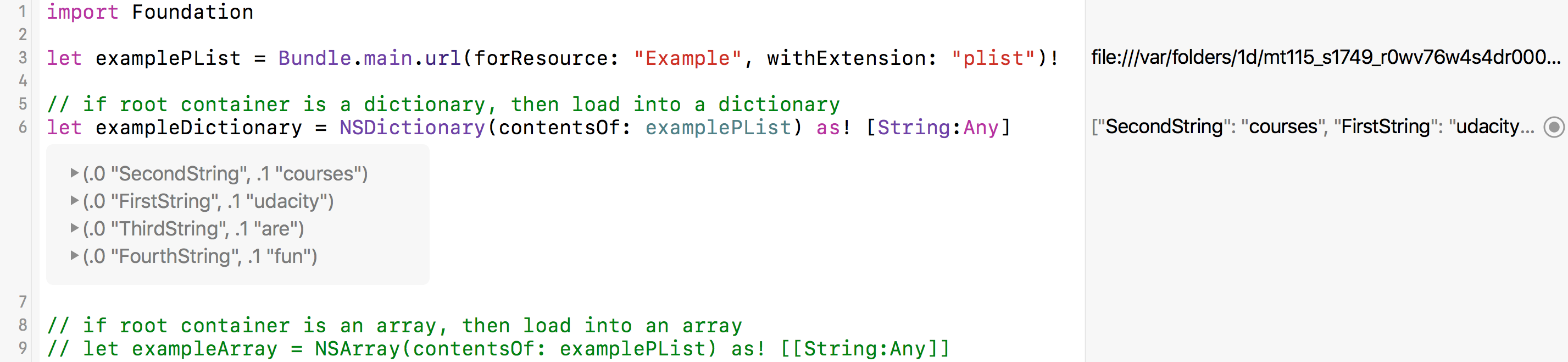
Nhưng, nếu Root container là một Dictionary, thì mỗi phần tử sẽ được xác định như một key duy nhất.



Khi này plist có thể được hiển thị với 3 cột: Key, Type, Value. Trong đó:

* Key:
* Type:
* Value:

Sau đó, chúng ta có thể load dữ liệu với một vài dòng code đơn giản như bên dưới:



* 1. Working with Plist Data in an Array
  2. Working with Plist Data in a Dictionary
  3. Using JSON Data

1. **Advanced Structs**
   1. Introduction To Advanced Structs
   2. Methods
   3. Custom Initializers
   4. Access Modifiers
   5. Public, internal and File Private
2. **Structs And protocols**
   1. Design with Protocols
   2. The ‘PlayingCard’ Protocol
   3. Extending an Existing Protocol
   4. Back to Our Deck Of Cards
3. **Classes**
   1. Gettings Started With Classes
   2. Value and Reference Types
4. **Classes and Inheritance**
   1. Introduction to Classes and Inheritance
   2. More About Inheritance
   3. Inheritance in Action
   4. The Super Keyword And Subclass Initializers
5. **Errors**
   1. Introduction to Errors
   2. Why Catch Errors?
   3. Handling Errors in Swift
   4. Guitarists and Broken Strings
   5. Optional Types and Errors
6. **Closures**
   1. Introduction to Closures
   2. Basic Closure Expression Systax
7. **Map and Reduce**
   1. Introduction to Map() and Reduce()
   2. Map()
   3. Reduce()

Swift for Developers:

1. **Swift Basics: Type, Variables and Constants**
   1. Intro to Swift Strings
   2. String interpolation
   3. The characters property
   4. String methods
   5. Swift types
   6. Constants andVariables
2. **Functions**
   1. Intro to Functions
   2. Global Functions and Methods
   3. Anatomy of a Function
   4. Externam and Local Parameter Names

Version Control with Git:

* 1. What is Version Control
  2. Create A Git Repo
  3. Review A Repo’s History
  4. Add Commits To A Repo
  5. Tagging, Branching and Merging
  6. Undoing Changes