*Hướng dẫn này được cập nhật dành cho bản Xcode 8 và Swift 3*

*Swift là ngôn ngữ dễ dàng để bắt đầu. Cú pháp thì đơn giản để hiểu và có xu hướng hồi tưởng lại các ngôn ngữ phổ biến khác mà bạn đã từng làm quen*

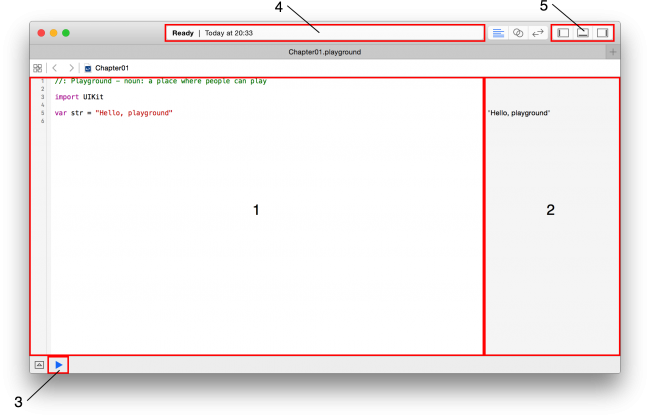
Bài 1. Làm quen với môi trường Playgrounds trong XCode

Apple đã giới thiệu Swift năm 2014 tại WWDC. Cùng thời gia này, công ty đã thêm Playgrounds tương tác trong XCode. Không còn nghi ngờ gì nữa Playground là cách tốt nhất để học ngôn ngữ Swift. Để tôi cho bạn thấy điều đó !

Mở XCode và tạo một **Playgrounds** mới bằng cách chọn **New** => **Playground** từ **File menu.** Sau đó nhập tên của playground, thiết lập **Platform** thành **iOS**. Nói cho XCode nơi bạn muốn lưu Playground và chọn **Create**



**Playground** được chia là 5 phần.



1. Source editor: Đây là khu vuecj mà bạn sẽ viết code Swift của mình. Nó giống như TextEdit hay Nodepad. Khi bạn tạo 1 Playground thì Xcode đã tạo sẵn cho bạn với 3 dòng code, 1 dòng comment nằm trên cùng, dưới nó là 1 câu lệnh import, và khai báo 1 biến.
2. Results sidebar: Khu vực hiển thị những kết quả của mã lệnh của bạn.
3. Execution control: Playground thực hiện tự động theo mặc định, có nghĩa là bạn có thể viết mã và ngay lập tức nhìn thấy đầu ra. Control này cho phép bạn thực hiện lại playground và thực hiện đầu ra là các câu lệnh print()
4. Activity viewer: Điều này cho thấy trạng thái của playground, nó cho thấy playground đã thực hiện xong hay sãn sang để xử lý. Khi playground đang thực hiện thì 1 indicate sẽ xoay
5. Panel controls: Thanh chia màn hình



Điều gì làm cho Playground trở nên tuyệt vời? Một Playground chứa một **Swift REPL**. REPL viết tắt của **Real-Eval-Print-loop**. Có nghĩa là Swift REPL đọc nội dung của playground, sau đó đánh giá code và hiển thị đầu ra trong bảng điều khiển kết quả. Quá trình xử lý này lặp đi lặp lại cho mỗi thay đổi bạn tạo ra trong trình soạn code.

Để thấy được điều đó. Thay đổi giá trị của biến str từ “Hello, playground” thành “Hello, word” và xem kết quả tự động thay đổi ở bảng kết quả bên phải.



**Comments**

Playground hiện tại chỉ chứa 3 dòng code, nhưng có một số điều chúng ta cần tìm hiểu thêm về ngôn ngữ lập trình Swift. Ví dụ, comments 1 dòng đơn bắt đầu bằng //

Swift cũng hỗ trợ comment nhiều dòng. Chúng bắt đầu với /\* và kết thúc \*/. Hãy thử nó bằng cách thêm chúng vào Playground

/\*

This is a multi-line comment.

It doesn't appear in the results panel.

\*/

Cũng như các ngôn ngữ lập trình khác trình dịch sẽ bỏ qua các comment, Swift cũng vậy. Những comment sẽ không hiển thị trong bảng kết quả bên phải.

**Modules**

Với dòng code thứ là là câu lệnh import framework UIKit. Theo sau từ khóa import là tên của module, Framework trong ví dụ này là UIKit.

import UIKit

Nhưng modules là gì? Trong Swift là một đơn vị phân phối mã. Điều này có thể là một thư viện hoặc một framework, nhưng một ứng dụng cũng có thể xem xét là một module.

Bằng cách import framework UIKit chúng ta được quyền truy cập vào các class, protocol, và hằng số được định nghĩa trong framework. XCode thêm câu lệnh import dành cho framework UIKit bởi vì chúng ta chọn nền tảng (platform) iOS khi chúng ta chọn iOS như một platform khi tạo playground. Trên iOS, các ứng dụng phụ thuộc rất nhiều vào UIKit và đó là lý do Xcode thêm câu lệnh import cho chúng ta.

Variables and Constans(Biến và hằng số)

Trên dòng code thứ 3, chúng ta khai báo một biến str có kiểu là String. Trong Swift, giá trị có thể lưu trữ như 1 biến hoặc 1 hằng số. Một biến được khai báo với từ khóa var còn hằng số sẽ được khai báo với từ khóa let.

Bạn có thể thay đổi biến str thành hằng số bằng cách thay thế từ khóa var bằng từ khóa let. Kết quả bên bảng kết quả sẽ không bị thay đổi.

let str = "Hello, world"

Chúng ta sẽ được tìm hiểu rõ hơn về biến và hằng số hay var và let ở bài sau!

**Semicolons**(Dấu chấm phẩy)

Có một chút gì đó không đúng từ đầu đến giờ? Đó là cuối mỗi câu lệnh chúng ta không sử dụng dấy chấm phẩy mà trình biên dịch không báo lỗi. Đó là sự khác biệt của Swift với các ngôn ngữ khác (Java, C/C++ …). Trình dịch của Swift đủ thông minh để hiểu được điều đó. Nếu bạn có sử dụng dấu chấm phẩy cuối câu lệnh cũng không sao nhưng điều đó không cần thiết.

*Tiếp theo: Chúng ta sẽ tìm hiểu về biến và hằng >>>*