



Ngôn ngữ Swift căn bản

Cài đặt và lập trình căn bản

Optional là gì?

- Optional là một khái niệm mới trong ngôn ngữ lập trình Swift
- Với việc sử dụng optional, ngôn ngữ Swift được xem là ngôn ngữ “an toàn” hơn so với ngôn ngữ Objective-C trước đó

// khai báo optional

var <tên biến>: <Kiểu dữ liệu>?


- Ví dụ:

// Khai báo Integer Optional

var perhapsInt: **Int**?

// Khai báo String Optional

var perhapsStr: **String**?

- 
- Trong Swift, khi khởi tạo các biến, các biến này mặc định sẽ được khởi tạo dưới dạng non-optional, tức là phải được gán giá trị mà không được để nil. Nếu chúng ta gán giá trị nil cho các biến non-optional này, trình biên dịch sẽ thông báo lỗi.

Ví dụ:

```
var str: String //compile error
```

// biến str là kiểu String, kiểu non-optional nhưng không gán giá trị mặc định nên Xcode sẽ báo lỗi

```
var str: String = "Hello Swift" // OK
```

```
str = nil // compile error
```

// vì str là biến non-optional, nên không thể gán cho nó bằng nil

Forced Unwrapping

Nếu có một biến được khai báo là optional và chúng ta muốn sử dụng giá trị của biến đó thì chúng ta phải unwrap biến đó

Ví dụ:

```
/* Nếu không unwrap*/  
  
// khai báo biến myString là một biến optional  
var myString:String?  
  
// khởi tạo giá trị cho biến myString  
myString = "Hello, Swift!"  
  
if myString != nil {  
    print(myString)  
} else {  
    print("myString has nil value")  
}
```

Kết quả trả về

```
Optional("Hello, Swift!")  
Program ended with exit code: 0
```

Force Unwrapping

Force unwrap bằng cách thêm ! sau biến cần unwrap. Tuy nhiên việc force unwrap là việc nên tránh, vì trường hợp nếu biến force unwrap đó nil thì sẽ gây crash app

Ví dụ:

```
/* Nếu unwrap*/  
// khai báo biến myString là một biến optional  
var myString:String?  
  
// khởi tạo giá trị cho biến myString  
myString = "Hello, Swift!"  
  
if myString != nil {  
    print(myString!)  
} else {  
    print("myString has nil value")  
}
```

Kết quả trả về

```
Hello, Swift!  
Program ended with exit code: 0
```

Automatic Unwrapping

```
var myString:String!
```

```
myString = "Hello, Swift!"
```

```
if myString != nil {  
    print(myString)  
} else {  
    print("myString has nil value")  
}
```

- Ở đây biến myString được khai báo kiểu String!, khi ta gọi hàm print (myString) thì biến myString này đã tự động unwrap thành kiểu String

Optional Binding

- Sử dụng optional binding để check xem biến optional có giá trị hay ko, để có những xử lý tương ứng tránh bị crash app
- Dùng **if let** hoặc **guard let** để binding optional

```
var myString:String?
```

```
myString = "Hello, Swift!"
```

```
if let yourString = myString {  
    print("\(yourString)")  
} else {  
    print("Your string does not have a value")  
}
```

Kết quả:

```
Hello, Swift!
```

```
Program ended with exit code: 0
```

if let và guard let

```
func optionalBinding() {  
    let name: String? = nil  
    let age: Int? = 3  
  
    if let ten = name{ // Khởi tạo một đối tượng ten = đối tượng optional name  
        print(ten) // sử dụng biến non-optional ten  
    }  
  
    // Khởi tạo một đối tượng tuoi = đối tượng optional age  
    guard let tuoi = age else {  
        print("age nil") // xử lý nếu age nil  
        return  
    }  
    print(tuoi) // sử dụng biến non-optional tuổi  
}  
  
optionalBinding()
```