

**すごいぞ!**

**新プログラミング言語 Swift**

**使ったことある？**

# Swiftとは？



**出た時期** 2014年6月

**特徴**

- モダン
- 高速
- 安全
- インタラクティブ

**Swiftは日本語に訳すと「迅速な」という意味。  
アイコンには「アマツバメ」をモチーフに  
している。**

え?なにこれ・・・すっごい

# breakが**不要**に switch文

読んで字の如し。C言語のswitchと同じ動作にしたい場合は「**fallthrough**」文を使用する。

# タプルがいい感じ!

複数個のデータを組みにしてまとめたもの。  
簡単なデータ受け渡し用途を想定。つまり  
こんなことができます。

```
func BMI(~~~~)
{
  let index = ~~~~~~
  let ideal = ~~~~~

  ~~~~~~

  ~~~~return (index, ideal)
}
```

```
let result = BMI(~~~)
~~あとはresultを好きなように使う~~

println(result.0)
println(result.1) とかね。
```

# 暗黙のキャストが一切だめ

```
var i8 : Int8 = 1
var i16 : Int16 = 1

i8 = i16 // コンパイルエラー
i16 = i8 // コンパイルエラー
```

煩わしい点がモロモロあるけど、結果的には  
こっちの方がバグが出にくくていいと思う

# 便利だな **範囲演算子**

「..**<**」とか「..**<**」とか。for文とswitch。文と、いろんな局面で使われます。

```
for i in 1..<5      // これは for i = 0; i<5; i++ と等しい
```

```
var ints1 = [10,20,30]
```

```
var ints2 = [40,50]
```

```
ints1 = ints1 + ints2           // [10,20,30,40,50]
```

```
ints2[1] = 55                 // [40,55]
```

```
ints1[1...2] = ints2         // [10,40,55,40,50]
```

# 小難しいぞ **optional!**①

平たく言うとnull(Swiftではnilという)アクセスを極力させない機構。通常、変数にはnilを入れることができない。

変数をOptional型として宣言して初めてnilを入れることが可能になる。

```
var abc:Int = nil //× Int型  
var def:Int? = nil //○ オptionalInt型
```

# 煩わしいぞ optional②

```
abc = def // × 代入不可能
```

なんと、オプション型は別の型として扱われるのでそのままでは代入できない!

```
abc = def! // ○ アンラップする
```

こうやってアンラップということをする必要がある。

# それでもいいかも **Optional**③

めんどくさくて、煩わしいけど……

**厳密なルールを課すことにより、ク  
ラッシュしにくいアプリケーション  
を作ることができるから  
まあ、いいかも。**

# 時にはいいかも **Extension**

```
extension String {  
    var lenght: Int {  
        return countElements(self)  
    }  
}
```

```
println("小惑星".lenght) // こんな感じね
```

**Swift はテーマ通り**  
**「モダン」「安全」「高速」**  
**「インタラクティブ」だった!**

**ご清聴ありがとうございました。**