

SI4 TP3 - Swift

Petits exercices pour découvrir Swift à travers quelques outils. C'est juste une découverte ...

Pour cela vous devez installer un environnement de développement qui permet de compiler Swift 1.2 :

Au choix :

- Avec [Visual Studio 2015 community Edition](#) + [Silver](#) : il est préférable d'avoir Windows 10.

Sinon vous pouvez travailler en ligne à travers un navigateur Web :

- Swift online with [SwiftStub](#)

ATTENTION : Il faut que vous puissiez au plus vite avoir à disposition une version Yosemite d'OSX avec Xcode en 6.x ou 7.x. C'est à dire la semaine prochaine ! Vous pouvez télécharger une image pour [virtualBox](#) ou [VMware player](#) sur vos peer-to-peer préférés.

Partie 1 :

Utilisation du langage Swift pour les deux exercices du TD1 que vous avez déjà commencé.

A partir du moment où vous avez les outils pour travailler, le code du premier exercice du TD1 suit sur la page suivante.

L'algorithme a été vu en cours !

Pour le second exercice vous devez imaginer l'algorithme et écrire le code de l'application en Swift.

Indice : Il y a beaucoup de chance que l'algorithme du second exercice ressemble beaucoup au premier

Pour Xcode 6.x + Swift 1.2 :

```
import Foundation

func input() -> NSString {
    var keyboard = NSFileHandle.fileHandleWithStandardInput()
    var inputData = keyboard.availableData
    return NSString(data: inputData, encoding:NSUTF8StringEncoding)!
}

var table : Int? = Int()
var signe : String

var inputStr : String

print("Table:")
inputStr = input() as String

//inputStr = inputStr.stringByTrimmingCharactersInSet("\n")
let output = (inputStr as NSString).stringByTrimmingCharactersInSet(NSCharacterSet.whitespaceAndNewlineCharacterSet())

table = output.toInt()!

print("Signe:")
inputStr = input() as String

let output2 = (inputStr as NSString).stringByTrimmingCharactersInSet(NSCharacterSet.whitespaceAndNewlineCharacterSet())

signe = output2
for var i : Int = 0 ; i < 10; i++
{
    if (signe == "+")
    {
        println("\(i) \(signe) \(table!) = \(i + table!)")
    }
    if (signe == "-")
    {
        println("\(i) \(signe) \(table!) = \(i - table!)")
    }
    if (signe == "*")
    {
        println("\(i) \(signe) \(table!) = \(i * table!)")
    }
    if (signe == "/")
    {
        println("\(i) \(signe) \(table!) = \(i / table!)")
    }
}
```

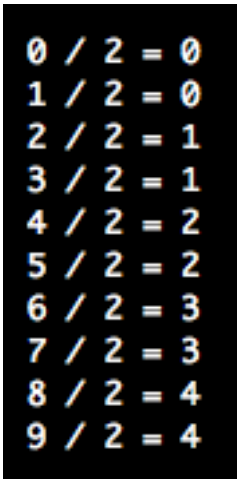
Résultat de l'exécution :

```
Table:2
Signe:+
0 + 2 = 2
1 + 2 = 3
2 + 2 = 4
3 + 2 = 5
4 + 2 = 6
5 + 2 = 7
6 + 2 = 8
7 + 2 = 9
8 + 2 = 10
9 + 2 = 11
|
```

Pour SwiftStub + Swift 1.2 :

```
3  var table : Int? = Int()
4  var signe : String
5
6  // pas de possibilité de faire une saisie clavier !
7  table = 2
8  signe = "/"
9
10 for var i : Int = 0 ; i < 10; i++
11 {
12     if (signe == "+")
13     {
14         println("\(i) \(signe) \(table!) = \(i + table!)")
15     }
16     if (signe == "-")
17     {
18         println("\(i) \(signe) \(table!) = \(i - table!)")
19     }
20     if (signe == "*")
21     {
22         println("\(i) \(signe) \(table!) = \(i * table!)")
23     }
24     if (signe == "/")
25     {
26         println("\(i) \(signe) \(table!) = \(i / table!)")
27     }
28 }
```

Résultat de l'exécution :



```
0 / 2 = 0
1 / 2 = 0
2 / 2 = 1
3 / 2 = 1
4 / 2 = 2
5 / 2 = 2
6 / 2 = 3
7 / 2 = 3
8 / 2 = 4
9 / 2 = 4
```

Partie 2 :

Cette partie ne porte que sur l'exercice 1 du TD1.

Modification de l'exercice 1. Vous allez devoir utiliser une fonction.

```
calcul(table!, signe)

//for var i : Int = 0 ; i < 10; i++
//{
//    if (signe == "+")
//    {
//        println("\(i) \(signe) \(table!) = \(i + table!)")
//    }
//    if (signe == "-")
//    {
//        println("\(i) \(signe) \(table!) = \(i - table!)")
//    }
//    if (signe == "*")
//    {
//        println("\(i) \(signe) \(table!) = \(i * table!)")
//    }
//    if (signe == "/")
//    {
//        println("\(i) \(signe) \(table!) = \(i / table!)")
//    }
//}
```

La fonction « calcul » remplace, les instructions en commentaire (vert). Vous devez écrire, le code de la fonction « calcul ».

Voici l'entête de la fonction « calcul » :

```
func calcul(table : Int, signe : String)
{
```

Le code de la fonction « calcul » est destiné à faire le calcul qui est passé en commentaire.

Partie 3 :

Manipulation de chaînes de caractères.

1. Créer deux variables, mettre « Bonjour » dans une, « Au Revoir » dans l'autre
2. Inverser les contenus des deux variables
3. Concaténer le contenu des deux chaînes dans une troisième (il va falloir chercher)
4. Insérer « et » entre « Bonjour » et « Au Revoir », dans une variable pour avoir : « Bonjour et Au Revoir ». Même principe que la question 3.
5. Supprimer les blancs dans la chaîne « Bonjour et Au Revoir ». Une petite recherche avec Google ... que vous devrez beaucoup utiliser en SLAM.