

Premiers pas avec Python, Scala et Swift

ACT – Algorithms and Computational Thinking

Résumé

Les exercices suivants, vous permettront de lancer vos premiers projets et d'écrire vos premières lignes de code avec les langages Python, Scala et Swift.

Table des matières

1	Introduction	3
2	Python	3
2.1	Afficher un simple message à l'aide de la fonction <i>print()</i>	3
2.2	Utiliser vos premières variables	4
2.3	Vos premiers calculs à l'aide de variables	4
2.4	Utilisation de commentaires	4
3	Scala	5
3.1	Afficher un simple message à l'aide de la fonction <i>println()</i>	5
3.2	Utiliser vos premières variables	5
3.3	Vos premiers calculs à l'aide de variables	5
3.4	Utilisation de commentaires	5
4	Swift	6
4.1	Afficher un simple message à l'aide de la fonction <i>print()</i>	6
4.2	Ajouter un label et afficher du text	6
5	Solutions	8
5.1	Python	8
5.2	Scala	9
5.3	Swift	10

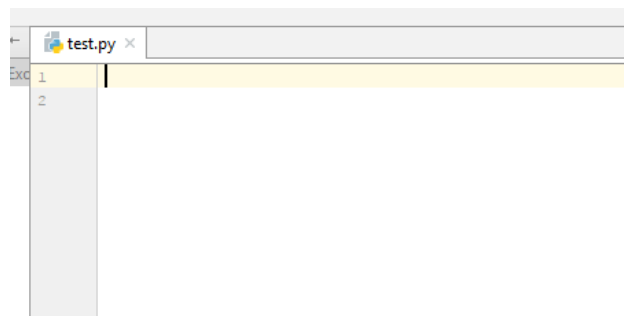
1 Introduction

Après avoir suivi les différents tutoriels d'installation pour *IntelliJ IDEA* (Python et Scala), nous vous proposons de lancer vos premiers projets, un pour chacun des langages (Python, Scala et Swift).

2 Python

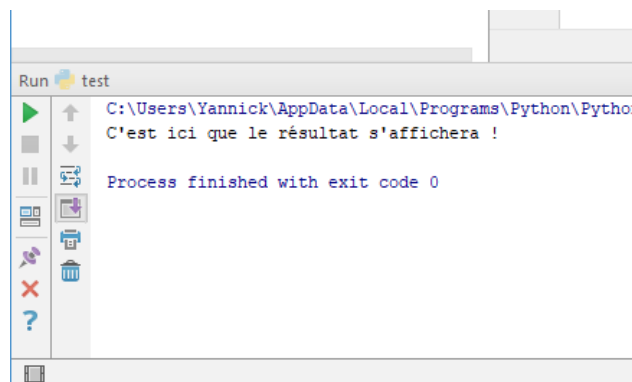
Commencez par ouvrir *IntelliJ* et créez un nouveau projet Python que vous nommerez *PythonTest*. Puis créez-y un nouveau fichier Python que vous nommerez *test.py* (c'est dans ce nouveau fichier que vous écrirez votre code).

Ecrivez vos lignes de code ici



Après avoir écrit un bout de code, il vous suffira de *Run* votre fichier pour le tester (cliquez droit sur votre fichier, puis sélectionnez *Run*).

Le résultat (l'output) se affichera dans la console en bas



2.1 Afficher un simple message à l'aide de la fonction *print()*

Essayez d'afficher : *ACT is great !*

Exemple : `print("votre_texte")`, `print(7)`, `print(nom_de_la_variable)`

2.2 Utiliser vos premières variables

Affectez la chaîne de caractères *"ACT is great !"* à la variable *t*, puis affichez là.
Affectez la valeur 4 à la variable *x*, puis affichez là.
Affectez la valeur 7 à la variable *y*, puis affichez là.
Note : *nom_de_la_variable = valeur* et n'oubliez pas *print()* permet d'afficher.

2.3 Vos premiers calculs à l'aide de variables

Affectez la multiplication de *x* et *y* à la variable *z*, puis affichez là.

2.4 Utilisation de commentaires

En ajoutant *#* au début d'une ligne de votre code, cela permet de spécifier à la machine que la ligne n'est qu'un commentaire et ne doit pas être prise en compte.

Essayez : `#print("Je ne m'affiche pas : ")`

Mais si vous voulez mettre plusieurs lignes en commentaire, vous pouvez utiliser des commentaires en bloc `"""vos_commentaires_sur_plusieurs_lignes"""` ou `'''vos_commentaires_sur_plusieurs_lignes'''` (Plus pratique que d'écrire à chaque ligne *#*).

Essayez :

```
'''print("Je ne m'affiche pas!")
l = 6
print(l)
Je suis un commentaire
'''
```

3 Scala

Commencez par ouvrir *IntelliJ* et créez un nouveau projet Scala que vous nommerez *ScalaTest*. Puis créez-y un nouveau fichier Scala de type object que vous nommerez *test* (c'est dans ce nouveau fichier que vous écrirez votre code).

De la même manière que pour Python vous pouvez *Run* votre code et voir le résultat.

3.1 Afficher un simple message à l'aide de la fonction *println()*

Essayez d'afficher : *ACT is great !*

Exemple : `println("votre_text")`, `println(7)`, `println(nom_de_la_variable)`

3.2 Utiliser vos premières variables

Affectez la chaîne de caractères "*ACT is great !*" à la variable *t*, puis affichez là.

Affectez la valeur 4 à la variable *x*, puis affichez là.

Affectez la valeur 7 à la variable *y*, puis affichez là.

Aide : `var nom_de_la_variable = valeur` et n'oubliez pas `println()` permet d'afficher

3.3 Vos premiers calculs à l'aide de variables

Affectez la multiplication de *x* et *y* à la variable *z*, puis affichez là.

3.4 Utilisation de commentaires

En ajoutant `#` au début d'une ligne de votre code, cela permet de spécifier à la machine que la ligne n'est qu'un commentaire et ne doit pas être prise en compte. Essayer : `//print("Je ne m'affiche pas : ")`

Mais si vous voulez mettre plusieurs lignes en commentaire, vous pouvez utiliser des commentaires en bloc `/* vos_commentaires_sur_plusieurs_lignes */` (Plus pratique que d'écrire à chaque ligne `//`).

Essayer :

```
/* print("Je ne m'affiche pas!")
l = 6
print(l)
Je suis un commentaire
*/
```

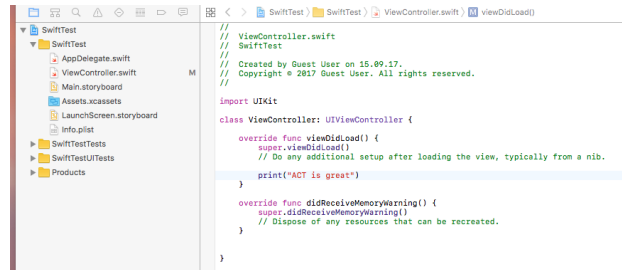
4 Swift

Commencez par ouvrir *Xcode* et créez un nouveau projet Swift (pour Iphone si possible) que vous nommerez *SwiftTest*.

4.1 Afficher un simple message à l'aide de la fonction *print()*

Allez dans le fichier *ViewController*.

Dans la méthode *viewDidLoad()* ajoutez cette simple ligne *print("ACT is great")*



```

//
//  ViewController.swift
//  SwiftTest
//
//  Created by Guest User on 15.09.17.
//  Copyright © 2017 Guest User. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
        print("ACT is great")
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

}

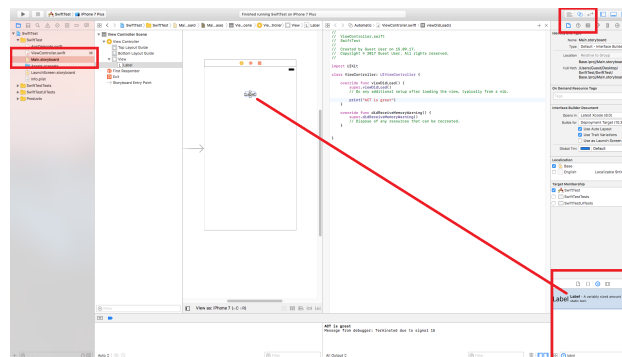
```

Puis runner le code, un émulateur Iphone devrait s'afficher avec une page vide après chargement. Mais dans la console de *Xcode* en bas vous devriez voir apparaître votre texte.

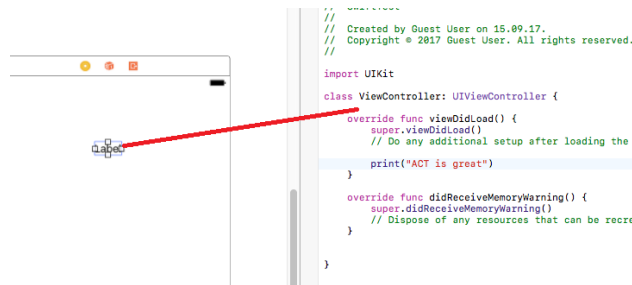
4.2 Ajouter un label et afficher du text

Maintenant nous allons essayer d'afficher du texte sur l'émulateur Iphone (et non simplement dans la console). Suivez simplement les étapes.

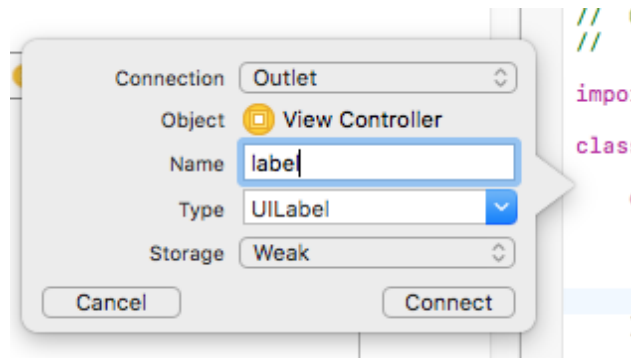
Ajoutez un label sur votre *Main.storybord*



Liez votre label à votre fichier *ViewController* en tirant votre label tout en appuyant sur Ctrl



Nommez votre label



Ajoutez cette ligne dans votre *ViewController* et lancer l'application

```

override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after load

    print("ACT is great")
    self.label.text = "ACT is great"
}

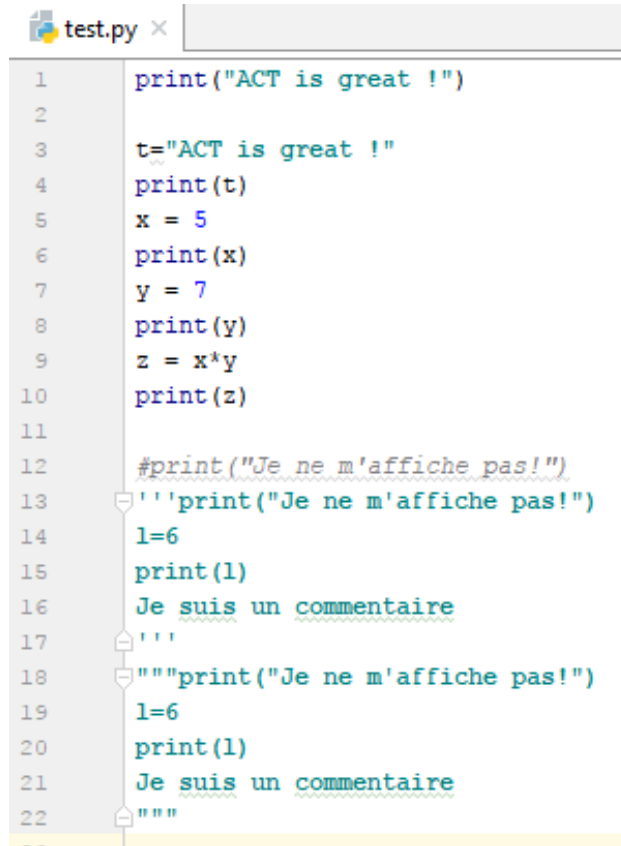
```

Vous devriez obtenir *ACT is great* dans la console et sur l'émulateur Iphone.

5 Solutions

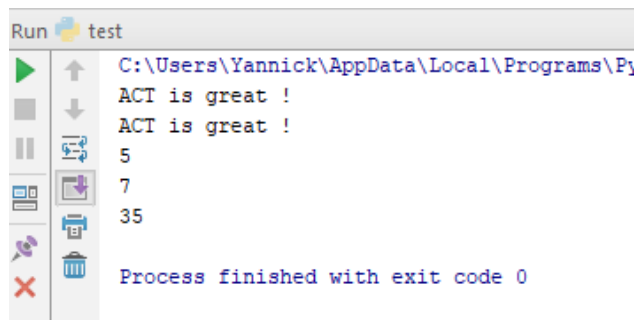
5.1 Python

Le contenu de votre fichier devrait ressembler à ça :



```
1 print("ACT is great !")
2
3 t="ACT is great !"
4 print(t)
5 x = 5
6 print(x)
7 y = 7
8 print(y)
9 z = x*y
10 print(z)
11
12 #print("Je ne m'affiche pas!")
13 '''print("Je ne m'affiche pas!")
14 l=6
15 print(l)
16 Je suis un commentaire
17 '''
18 """print("Je ne m'affiche pas!")
19 l=6
20 print(l)
21 Je suis un commentaire
22 """
```

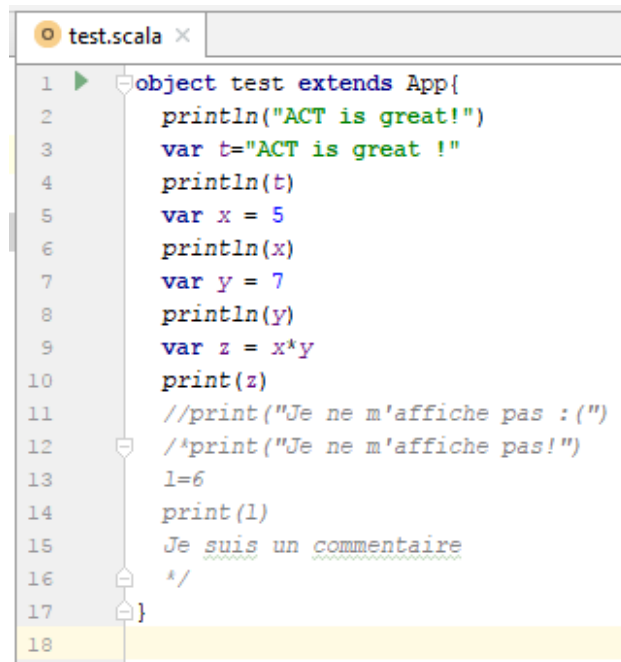
Et la console devrais vous retourner qqch comme ça :



```
Run test
C:\Users\Yannick\AppData\Local\Programs\Py
ACT is great !
ACT is great !
5
7
35
Process finished with exit code 0
```

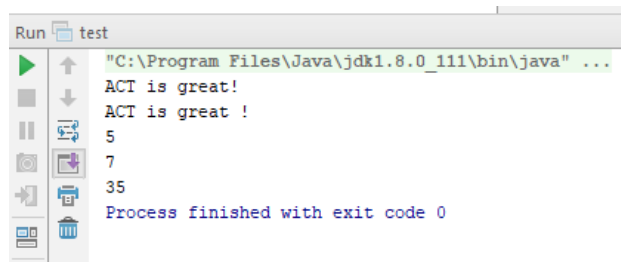

5.2 Scala

Le contenu de votre fichier devrait ressembler à ça :



```
1 object test extends App{
2   println("ACT is great!")
3   var t="ACT is great !"
4   println(t)
5   var x = 5
6   println(x)
7   var y = 7
8   println(y)
9   var z = x*y
10  print(z)
11  //print("Je ne m'affiche pas :")
12  /*print("Je ne m'affiche pas!")
13  l=6
14  print(l)
15  Je suis un commentaire
16  */
17 }
18
```

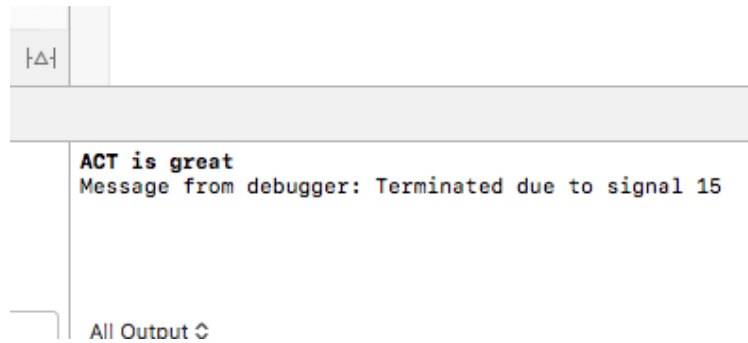
Et la console devrais vous retourner qqch comme ça :



```
Run test
"C:\Program Files\Java\jdk1.8.0_111\bin\java" ...
ACT is great!
ACT is great !
5
7
35
Process finished with exit code 0
```

5.3 Swift

La console de votre projet Swift devrait ressembler à ça après avoir Run :



Et votre émulateur à ça à la fin :

