

Chapitre 4 : Les variables

I. Créer une variable avec scala

En scala, nous avons deux mots clés nous permettant de déclarer une variable

- **var** : lorsqu'on utilise ce terme, nous avons la possibilité de modifier la valeur de la variable autant de fois que l'on veut on dit alors qu'elle est mutable.
- **val** : une variable créée avec val n'est pas modifiable

Essayons de créer une variable de type double et lui affecter une valeur entière et inversement.

```
scala> var note:Double = 10_
note: Double = 10.0

scala> var note_1:Int = 12.3
<console>:11: error: type mismatch;
 found   : Double(12.3)
 required: Int
    var note_1:Int = 12.3
                        ^
```

Dans le premier cas, on constate que la variable **note** est créée sans problème car **int** est un type de données plus petit et **double** est un type de données plus grand. Ainsi la conversion est automatiquement faite.

Par contre l'inverse est impossible également à cause du fait que le type de donnée **double** est beaucoup plus grand que **int**.

II. Passer une variable par valeur ou par référence

```
scala> var x:Int = 10
x: Int = 10

scala> var y=x
y: Int = 10

scala> x=66
x: Int = 66

scala> y
res0: Int = 10

scala>
```

Dans l'exemple ci-dessous, on crée 2 variables **x** et **y** ensuite :

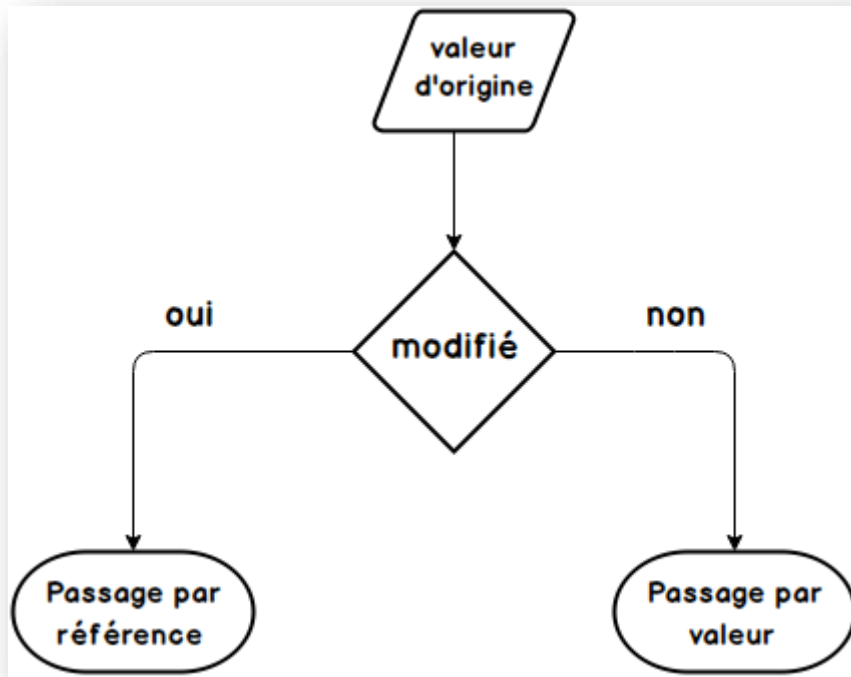
- On affecte respectivement **10** à **x** et **x** à **y**

Chapitre 4 : Les variables

- On change la valeur de **x** à 66

Après cela on constate que la valeur de **x** a bien changé alors que **y** reste à 10.

Pour expliquer cela intéressons-nous aux concepts de **passage par valeur** et **passage par référence**



Dans notre cas, c'est la valeur actuelle de la variable **x** est assigné à **y** on parle alors de **passage par valeur**.

Dans le cas où la variable **y** avait changé de valeur, on parle dès lors de **passage par référence** dans ce cas alors c'est l'emplacement adresse de **x** qui sera transmis à **y** « toute modification apportée aux arguments formels se reflétera également dans les arguments réels. »

	Passage par valeur	Passage par référence
variable	Une copie de la variable est transmise	La variable elle-même est transmise
effet	La modification de la copie de variable n'entraîne pas la modification de la valeur d'origine	La modification de la variable affecte également la valeur de la variable copié

Chapitre 4 : Les variables

III. Créer plusieurs variables sur une seule ligne

```
scala> var x=10;var y=11;var z=12;  
x: Int = 10  
y: Int = 11  
z: Int = 12  
  
scala>
```

Pour créer plusieurs variables sur une seule ligne, chaque déclaration de variable doit être séparée par un point-virgule.