Table des matières

[**I.** **Créer une variable avec scala** 2](#_Toc117939659)

[**II.** **Passer une variable par valeur ou par référence** 2](#_Toc117939660)

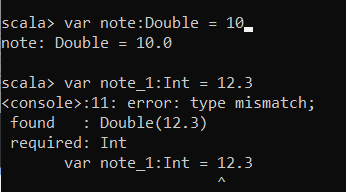
[**III.** **Créer plusieurs variables sur une seule ligne** 4](#_Toc117939661)

# **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

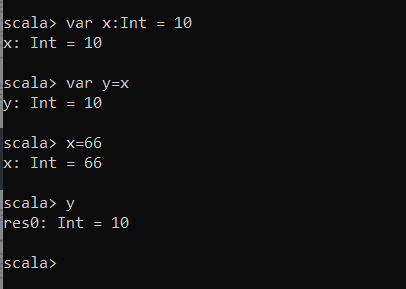
Essayons de créer une variable de type double et lui affecter une valeur entière et inversement.



Dans le premier cas, on constate que la variable **note** est créer 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**.

# **Passer une variable par valeur ou par référence**

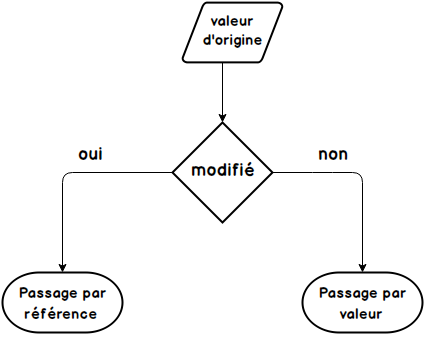


Dans l’exemple ci-dessous, on crée 2 variables **x** et **y** ensuite :

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

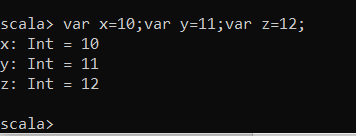
****

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’entraine pas la modification de la valeur d’origine | La modification de la variable affecte également la valeur de la variable copié |

# **Créer plusieurs variables sur une seule ligne**



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