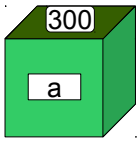
TP1 Python : Affectations, entrée, sortie

Dans les programmes, on doit souvent stocker des données pour les utiliser plus tard. On utilise pour ça des variables. Supposons que l'on veuille stocker la valeur 300, on peut utiliser une variable nommée par exemple « a », cette variable peut être vue comme une boîte :



En langage naturel, on peut écrire :	a prend la valeur 300
ou :	Affecter à a la valeur 300
En pseudo-code, on écrira ceci :	300 → a (on met 300 dans la boîte « a »)
ou ceci :	a ← 300
En Python, on écrira cela :	a = 300

Remarques :

- Le signe = n'a pas le même sens en informatique et en mathématiques.
  Ainsi, en informatique, l'instruction « 300 = a » ne veut rien dire (on ne peut pas mettre a dans 300 !).
  De même, en informatique, « a = b » ne veut pas dire la même chose que « b = a » !
- L'instruction « a ← 300 » est une affectation (on affecte la valeur 300 à la variable « a »).

Exercice 1 : affectations

Traduisez en Python les algorithmes suivants tout en les entrant dans la console de Thonny et donnez la valeur finale de r.

1°) Algorithme écrit en langage naturel :

Affecter à m la valeur 2
Affecter à n la valeur m – 1
Affecter à p la valeur n + 4
Affecter à q la valeur 3 p
Affecter à r la valeur p – m

2°) Algorithme écrit en « pseudo-code » :

```
p ← 1
r ← p + 3
p ← p – 1
r ← p + 1
```

Remarques :

- le pseudo-code est une façon d'écrire des algorithmes ;
- un tableau de valeurs des variables permet de vérifier qu'un programme fait ce qu'on attend de lui.

Exercice 2 : affectations, (re)découverte de deux règles

Pour chaque question, tapez les instructions suivantes en mode console et dîtes ce que vous remarquez :

1°)
```
>>> a = 4
>>> a = 1
>>> a
```

2°)
```
>>> a = 4
>>> b = 7
>>> a = b
>>> a
>>> b
```

3°)
```
>>> a = 4
>>> b = 7
>>> b = a
>>> a
>>> b
```

Deux propriétés de l'affectation

Écrasement : quand on met une valeur dans une variable, toute valeur existant déjà dans la variable est effacée (écrasée).

Par exemple, à la suite des deux instructions :

a ← 2
a ← 3

la variable « a » contient 3 et pas 5.

Si on veut ajouter 3 au contenu précédent de « a », on doit faire :

a ← a + 3

Copie : l'affectation entre variables copie le contenu d'une variable sans l'effacer.

Par exemple, à la suite des trois instructions :

a ← 2
b ← 7
a ← b

le contenu de « b » est copié dans « a ». Ainsi les deux variables *a* et *b* contiennent la valeur 7.

Remarque : comme dit précédemment, « a = b » ne veut pas dire la même chose que « b = a ».
« a = b » veut dire : on met la valeur de b dans a ; « b = a » veut dire : on met la valeur de a dans b.