

Exercices : Algorithmique et programmation :

Algorithmes et pseudo-code Pour calculer 5^4 je peux faire $5 \times 5 \times 5 \times 5 = 625$ en faisant 3 multiplications, mais aussi $5 \times 5 = 25$ puis $25 \times 25 = 625$ en faisant 2 multiplications, et on ne peut pas faire mieux. Comment calculer 5^7 en faisant le moins de multiplications? Comment calculer a^n avec n entier en faisant le moins de multiplications possibles?

Ecrivons en français les deux méthodes de calcul vues ci-dessus :

1. Etant donné a un nombre
2. Calculer $a*a*a*a$

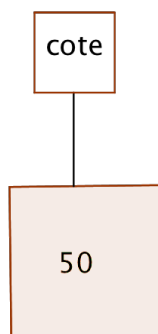
1. Etant donné a un nombre
2. Calculer $b = a*a$
3. Puis calculer $b*b$

On va exprimer ces suites d'opérations ou **algorithmes** dans un langage plus pratique appelé **pseudo-code**

```
a <- Entrer("Donnez un nombre")
p <- a*a*a*a
afficher(p)
```

Cet algorithme contient 3 lignes ou **instructions** :

1. Dans la première instruction a est une **variable**. On verra mieux plus loin à quoi correspond cette notion. Pour l'instant imaginons a comme un nom sur une boîte (comme un nom sur une boîte aux lettres) dans lequel on peut ranger des nombres. La **fonction** entrer() dit à l'utilisateur de l'algorithme de donner un nombre, ensuite ce nombre est mis dans la boîte associée à a ou encore affecté à a. Le symbole <- est le symbole d'**affectation**, montrant que le nombre donné par l'utilisateur est **associé** à la variable a



2. La deuxième instruction signifie plusieurs choses : d'abord on prend le nombre associé à a et on le multiplie avec lui-même et on associe le résultat à une autre variable p
3. La fonction afficher() révèle le contenu de p à l'utilisateur
4. Le contenu d'une variable change au gré des affectations successives

Ex n° 1 Ecrire en pseudo-code le deuxième algorithme de calcul de a^4

Ex n° 2 Décrire les différents états de la variable a après chaque instruction de l'algorithme suivant

```
a <- 2
a <- a + 1
a <- a - 1
```

Ex n° 3 Que fait l'algorithme suivant ?

```
a <- 2
b <- 3
c <- a
a <- b
b <- c
```

Traduction d'un algorithme en un programme Python Nous allons maintenant traduire les deux algorithmes précédents en Python qui est un langage de programmation.

Lancer IDLE, la première fenêtre qui est ouverte est celle de l'interpréteur Python, on dit que Python est un langage interprété, le curseur clignotant | en dessous des chevrons > > > signifie que l'interpréteur est prêt à faire exécuter par la machine vos instructions sous réserve qu'elles soient correctement écrites.

Tapez par exemple 5*5 et faites entrer vous voyez affiché 25

Vous pouvez même entrer une instruction plus complexe, cette instruction sera interprétée puis exécutée, mais ce n'est pas pratique d'exécuter un programme de la sorte. On va plutôt se servir d'un éditeur de texte

Dans la barre de menus sélectionner Nouveau Fichier dans Fichier. Une fenêtre d'éditeur de texte apparaît dans lequel on va écrire le programme suivant, qui est une traduction du premier algorithme :

```
nombre = int(input("Donnez un nombre entier "))
puissance = nombre*nombre*nombre*nombre
print(puissance)
```

Faire CTRL+S pour sauvegarder ce programme dans vos documents en le donnant un nom par exemple puissance4.py

Exécuter ce programme plusieurs fois en tapant F5 au clavier à chaque fois et donner différentes valeurs à nombre

Quelques remarques :

1. Quand vous entrez un nombre entier au clavier par exemple 5, ce n'est pas considéré comme un nombre , ainsi la fonction `int()` sert à transformer le 5 entré au clavier en 5 le nombre entier (integer signifie entier en anglais)
2. En Python le symbole d'affectation est `=` . **Attention à ne pas confondre avec le symbole `=` en mathématiques !**
3. Que se passe-t-il si on entre au clavier 2.5 ? si on entre Python ?
4. La fonction `input()` prend comme argument un texte écrit entre `"` ou `"` qui sera visible à l'écran
5. La fonction `print()` affiche à l'écran le contenu de la variable entre `()`

C'est quoi une variable au juste ? Ouvrir le navigateur Firefox et tapez dans la barre d'adresse directement sans passer par un moteur de recherche

<http://www.pythontutor.com/live.html#mode=edit>

Sélectionner Python 3.6 et écrire le premier programme dans la fenêtre

Exécuter le programme pas à pas et vous voyez dans la zone grisée Frames **deux étiquettes nombre et puissance associées à deux cases dans lesquelles se trouvent des nombres**

Dans IDLE , dans l'interpréteur exécutez la fonction `vars()` vous voyez entre `{}` ce qu'on appelle un dictionnaire qui associe à la variable nombre le dernier nombre entré

Exécutez `id(nombre)` dans l'interpréteur et vous voyez l'adresse mémoire de la variable nombre, vous conviendrez que c'est moins pratique que le nom nombre

Retenir qu'une variable est une étiquette plus pratique pour un humain qu'un nombre de la forme 4297370976 pour se référer à un endroit précis de la mémoire de la machine

Ex n°4 Traduire en Python le deuxième algorithme

Ex n°5 Que fait l'algorithme suivant ? Faire un tableau pour montrer l'évolution des variables a et p après chaque instruction

```
a <- Entrer("Donnez un nombre")
p <- a*a
p <- p*a
p <- p*p
p <- p*a
afficher(p)
```

Ex n°6 Traduire l'algorithme précédent en Python et mettre des `print()` entre chaque instruction pour voir l'évolution des variables

Ex n°7 La méthode binaire : Pour calculer a^n on "décompose" n en paquets de 2 comme dans l'exemple suivant :

$a^7 = a^{3 \times 2 + 1} = (a^{2 \times 1 + 1})^2 \times a = ((a^2) \times a)^2 \times a$ ce qui donne l'algorithme

```
a <- Entrer("Donnez un nombre")
p <- a*a
p <- p*a
p <- p*p
p <- p*a
afficher(p)
```

Donner l'algorithme par la méthode binaire pour x^{15} . Combien de multiplications ?

Ex n°8 La méthode des facteurs : On décompose $n = p \times q$ en facteurs premiers on prend le plus petit facteur premier par exemple $n = 15 = 3 \times 5$ on calcule $a^3 = y$ par la méthode binaire puis $a^{15} = (a^3)^5 = y^5$ on calcule y^5 par la méthode binaire aussi car pour $n = 3$ et $n = 5$ la méthode binaire minimise le nombre de multiplications. D'où l'algorithme

```
a <- Entrer("Donnez un nombre")
p <- a*a
y <- p*a
p <- y*y
p <- p*p
p <- p*y
afficher(p)
```

Pour calculer n^{33} que choisir la méthode binaire ou la méthode des facteurs ?