

SNT : initiation à la programmation (1)

Cette année nous utiliserons essentiellement le langage Python pour pouvoir programmer. Nous utiliserons plusieurs environnements pour cela : <https://basthon.fr/> sur les téléphones et sur certains ordinateurs du lycée, édupython quand les élèves auront reçu leur ordinateur personnel et l'éditeur Python à l'intérieur de la calculatrice.

La séance du 8 septembre s'est terminée par la présentation au tableau du programme ci-contre.

```
1  nbr_val=0
2  nombre = int(input("Quel est votre nombre ? "))
3  for i in range(1, nombre+1) :
4      if nombre%i==0 :
5          print(nombre, " est divisible par ", i)
6          nbr_val=nbr_val+1
7  if nbr_val==2 :
8      print(nombre, " est un nombre premier")
9  else :
10     print(nombre, " n'est pas un nombre premier")
```

Opérations numériques de base

+ - * et / serviront respectivement à faire des additions, soustractions, multiplications et divisions.

** servira à faire des puissances.

Exemple 5^2 s'écrira `5**2`

// donne le quotient d'une division euclidienne. Exemple `17//3` donne 5

% donne le reste d'une division euclidienne. Exemple `17%3` donne 2

Stockage

Pour garder en mémoire des valeurs intéressantes obtenues lors de nos calculs on les stockera dans des variables.

Dans le programme de fin de séance :

A la ligne 1, on a stocké 0 dans la variable `nbr_val`

A la ligne 6, dans la variable `nbr_val` on stocke l'ancienne valeur de la variable à laquelle on rajoute 1, ça veut dire qu'on augmente de 1 la valeur qu'on avait.

Test : If et else

On peut tester si une relation entre deux quantités est vraie. Un test est toujours suivi de « : » et les lignes qui seront exécutées si le test est positif seront décalées d'une tabulation vers la droite.

Exemple 1 : A la ligne 4, on teste si le reste de la division euclidienne de nombre par i vaut 0, autrement dit si i divise « nombre ». Les lignes 5 et 6 qui suivent sont décalées et donc elles ne seront exécutées que si la condition est vraie, sinon on passera directement à la ligne 7.

Exemple 2 : A la ligne 7 on regarde si le nombre de diviseur est 2, et si c'est le cas la ligne décalée 8 sera exécutée. A la ligne 9 on a « else : » ce qui nous permet de savoir quoi faire si la condition n'est pas réalisée : l'instruction proposée à la ligne 10.

Boucle For

Quand nous voulons répéter un nombre prédéterminé de fois une série d'opération nous utiliserons une boucle **for**

Range produit une série de valeurs comprises entre « déb » comprise et « fin » exclue.

```
for variable in range(deb, fin) :  
  
    instruction 1  
  
    instruction 2  
  
    ...
```

Interactions

Sortie : Afficher une information

On utilisera la fonction `print(partie1, partie2, partie3, ...)` qui nous permet d'écrire à la suite différentes informations : chaînes de caractères entre guillemets, valeurs, contenu de variables (on écrira juste leur nom). A la fin de la sortie l'ordinateur passera automatiquement à la ligne.

Exemple : à la ligne 5, sur la même ligne on affiche le contenu de la variable « nombre », une petite phrase et le contenu de la variable « i ».

Entrée : Récupérer une information

Pour interagir on peut poser des questions avec la commande « `input()` », la réponse proposée par l'utilisateur est prise par l'ordinateur comme une suite de caractère, si on a envie qu'il comprenne qu'on lui a donné un entier il faudra utiliser la commande `int()`.

Exemple : à la ligne 2, on veut que l'utilisateur nous donne un nombre entier, celui-ci sera stocké dans « nombre » pour pouvoir être utilisé ultérieurement.

SNT : initiation à la programmation (2)

Exercice 1

Ecrire un programme :

- 1) Ecrivant tous les nombre de 1 à 10
- 2) Ecrivant les multiples de 5
- 3) Ecrivant la table de 5 (1x5=5 2x5=10 ...)
- 4) Demandant un nombre, et en donnant la table de multiplication

Exercice 2

Ecrire un programme

- Demandant à l'utilisateur un dividende et un diviseur qui seront stockés dans des variables portant ces noms.
- Calculant le quotient et le reste
- Ecrivant sur l'écran sa réponse sous la forme :
dividende=quotient x diviseur + reste

Exercice 3 boucles imbriquées

Ecrire un programme donnant toutes les coordonnées des points d'abscisses et d'ordonnées positives mais inférieures à 10.

Indication si x vaut 5 et y vaut 7, la commande `print("(" ,x," ;",y,")"` provoquera l'affichage (5 ;7)

Point cours

Si lors d'un appel à la fonction on ajoute à la fin de la parenthèse ,end="chaîne de caractères" au lieu d'aller à la ligne l'ordinateur écrira la chaîne.

Exemple :

```
1 print(5, end=" ")
2 print(6, end="-")
3 print(7, end="oo")
4 print(8)
```

A pour effet :

```
>>> # script executed
5 6-7oo8
```

Exercice 4

Ecrire les programmes permettant d'avoir les sorties suivantes :

```
>>> # script executed
*****
*****
*****
*****
*****
*****
```

```
>>> # script executed
*
**
***
****
*****
*****
```

```
>>> # script executed
*****
*****
*****
*****
****
**
```

Improvisez d'autres formes

Boucle while

Parfois on voudra répéter une série d'action jusqu'à ce qu'une condition soit réalisée, ou uniquement tant qu'une condition est réalisée.

```
while condition :  
    instruction 1  
    instruction 2  
    ...
```

Exercice 5

```
1 valeur = 100  
2 while valeur > 0 :  
3     valeur = valeur - 1  
4     print(valeur)  
5     print("boom!!!")
```

voici un programme

- 1) Sans le tester indiquez ce qu'il va faire
- 2) A votre avis est ce que la valeur 0 sera atteinte
- 3) Testez le programme et vérifiez votre conjecture.

Exercice 6

En vous inspirant de l'exercice précédent

Utiliser la fonction while pour permettre l'affichage des suites de nombres suivants :

- 0, 5, 10, 15, ..., 50
- 243, 250, 257, ..., 320
- * 700, 699, 698, ... 600
- * -250, -260, -270, ..., -380

Exercice 7

Faire un programme permettant de déterminer la liste des facteurs permettant de faire une décomposition d'un nombre donné.

Indications :

- On stockera dans la variable « nombre » le nombre demandé par l'utilisateur. Le contenu de cette variable changera pendant l'exécution du programme.
- On va la diviser par tous les diviseurs trouvés jusqu'à ce qu'il ne reste plus que 1. Donc la grande boucle sera while
- On va tester sur « nombre » différents diviseurs potentiels, allant de 2 à
- dès qu'on trouvera un diviseur, on l'écrira sur l'écran, et on recommencera avec des diviseurs allant de 2 à

Exercice 8

On veut faire un programme permettant de déterminer le PGCD de deux nombres entiers. On suivra pour cela l'algorithme proposé en classe dans le polycopié de cours. Vous pourrez réutiliser ce que vous avez fait à l'exercice 2.