

Nom & Prénom : .....

## Programmation en python : Séance n°1 : Premiers programmes

### Tâche n°1 : Calculer avec python

- 1. Dans un la suite MCNL, sélectionner la matière SNT-NSI, et cliquer sur l'icône EduPython.



The screenshot shows the EduPython environment. The top part is a code editor with the following Python code:

```
t=10
while t>=0 :
    print(t)
    t=t-1
print("boum !!!!")
```

A red circle labeled '1' highlights the code area. The bottom part is a terminal window showing the output of the code execution:

```
File "C:\Users\User\AppData\Local\Programs\Python\Python313\Lib\tkinter\_init__.py", line 2063, in __call__
  def __call__(self, *args):
KeyboardInterrupt
>>> chaine1="bonjour"
>>> print(chaine1[:-1]
...
...
)
bonjou
>>>
```

A red circle labeled '2' highlights the terminal window.

### Vision de l'interface des environnements EduPython et Pyscripter

Dans la **zone 1** on peut écrire des programmes, qui seront effectué quand on appuie sur ➤ : la flèche verte dans la barre supérieure.

La **zone 2** est la console, c'est la zone d'exécutions de ce qu'on a dans la zone 1, mais on peut aussi taper des instructions à l'intérieur et quand on appuiera sur « entrée » l'instruction de la ligne en cours sera exécutée.

- 2. Voici comment tester les deux façons de calculer  $5^7$  avec python ci-dessous :

Directement dans la zone d'exécution ou console	Dans un fichier programme
<pre>5**7 puis « entrée » &gt;&gt;&gt; 5**7 78125</pre>	<p>L'exécution de <math>5^{**}7</math> avec ➤ ne provoque rien. Si on veut qu'un programme génère un affichage, il faudra lui dire de manière explicite :</p> <pre>print(5**7), provoquera l'affichage suivant : *** Console de processus distant Réinitialisée *** 78125</pre>

De la même manière déterminer  $125^{-3}$  : .....

En plus des quatre opérations habituelles et des puissances nous avons :

- $254//43$  le quotient de la division euclidienne de 254 par 43
- $254\%43$  le reste de la division euclidienne de 254 par 43
- $4.562E7$  correspond à  $4,562 \times 10^7$

► 3. En passant par la console ou par un programme déterminer quels sont le reste et le quotient de la division euclidienne de 457895 par 124. Quotient : ..... Reste : .....

► 4. Pour déterminer la racine carrée de 57 on devra utiliser une fonction de la librairie « math »:

Directement dans la console	Dans un fichier programme
<pre>&gt;&gt;&gt; from math import * &gt;&gt;&gt; print(sqrt(57)) 7.54983443527075</pre>	<pre>" " " " " " " " " &gt;&gt;&gt; from math import * &gt;&gt;&gt; print(sqrt(57))</pre>

### Exercice n°1 :

a) Calculer  $18^{18}$  puis  $18^{-18}$  et enfin calculer  $18^{18} \times 18^{-18}$ . Qu'avez-vous obtenu ?

.....

x=10	x=10	x=10
y=2	y=x+5	y=2*x
print(x+y)	print(x+2*y)	print(3*y)

b) Quel est le reste et le quotient de 123456788 par 17 ?

c) Transformer 55555 secondes en heures, minutes et secondes.

..... heures, .....minutes .....secondes

d) Deviner et écrire dans la case vide ce qu'affiche le script de chaque colonne (voir à droite) :

e) En important au préalable le module math avec l'instruction : `from math import *`.

Calculer  $\sqrt{25}$  .....puis  $\sqrt{3^2 + 5^2}$  .....

### Tâche n°2 : premiers programmes

#### Exercice 2 : divisions euclidiennes

copier / coller le code suivant dans la zone programme.

Pour copier on surligne, alors que l'on appuie sur « Ctrl », on appuie aussi brièvement sur « c »

Pour coller met le curseur là où l'on veut voir apparaitre le texte, alors que l'on appuie sur « Ctrl », on appuie aussi brièvement sur « v »

Une fois le collage effectué, compléter le programme et vérifier qu'il fonctionne (utiliser dividende : 23 et diviseur : 7 , il doit vous donner un quotient de 3 et un reste de 2)

```
dividende = int(input(" votre dividende est : "))
# input pose une question et récupère une information de l'utilisateur
# int la convertit en entier
# = la stocke dans un variable dividende c'est un espace mémoire possédant ce nom
# le symbole # sert à écrire des commentaires qui seront ignoré par python

# dans l'espace vide ci-dessus vous écrirez une ligne permettant de récupérer
# le diviseur choisi par l'utilisateur et de le stocker dans la variable de
# même nom. La ligne d'après sert à afficher et préciser le résultat obtenu.
quotient = dividende // diviseur
print("le quotient sera : ", quotient)
# écrire en dessous les lignes permettant de calculer, stocker, et afficher le
# reste de notre division euclidienne
```

Si vous ne pouvez copier-coller, recopiez les lignes en gras et complétez avec les trois lignes demandées.

Utiliser votre programme pour vérifier le travail fait à la question 3 de la tâche 1.

### Exercice3

► 1. Tapez ou copier/coller le programme ci-contre. Exécutez ce programme plusieurs fois avec des valeurs différentes pour a et b. A quoi sert ce programme ?

```
a = int (input("valeur de a : "))
b = int (input("valeur de b : "))
c= (a+b)/2
print(c)
```

► 2. Créez un programme qui calcule la moyenne de trois nombres.

#### Exercice n°3 :

Ecrire un programme qui :

- affecte la valeur 2 à la variable a
- affecte la valeur 3 à la variable b
- affiche la valeur de la somme de a et de b

#### Exercice n°4 :

Ecrire un programme qui :

- demande un nombre entier pour la variable a
- demande un nombre entier pour la variable b
- affiche la valeur du produit de a et de b

## Tâche n°3 : Instruction Conditionnelle

```
from random import *
cache = randint(1,100)
print(cache)
```

► 1. Tapez le programme ci-contre. Exécutez ce programme plusieurs fois. A quoi sert ce programme ?

► 2. Jouons à un jeu contre l'ordinateur : l'ordinateur choisit un nombre au hasard compris entre 1 et 100 et vous devez le deviner. Vous ne pouvez faire qu'une seule proposition et l'ordinateur répond par "trop grand", "trop petit" ou "gagné".

Compléter les le programme ci-contre exécute le jeu proposé.

```
from random import *
cache = randint(1,100)
proposition = int (input("votre proposition ?"))
if proposition == cache :
    print( ) # à compléter avec un message de félicitations
elif proposition > cache :
    print( ) # à compléter avec un message approprié
else :
    print( ) # à compléter avec un message approprié
```

Remarque : quand une ligne se termine par « : » la ou les suivantes sont décalées d'une tabulation (touche « Tab » complètement à gauche sur le clavier).

#### Exercice n°5 :

Ecrire un programme qui :

- demande votre note
- affiche si vous avez ou non la moyenne.

## Tâche n°4 : boucle

Pour répéter un certain nombre de fois une action on peut utiliser une boucle for comme celle qui est proposé ci-contre. Recopier ou copier/coller là pour voir ce que cela produit.

Combien de fois le programme a-t-il écrit « ah » dans la console ?

Que se passe-t-il si on décale la dernière ligne avec une tabulation ?

```
for i in range(10):
    print("ah")
    print("tchoum")
```

Recopier ou copier/coller le programme ci-contre. Tester le.  
range(10) crée une sorte de liste allant de 0 (inclus) à 10 (exclus) allant de 1 en 1 et qui sera parcourue par la variable i.

```
for i in range(0,10,1):
    print("numero d'itération : ",i)
```

**Exercice n°6 :** Ecrire un programme qui affiche tous les nombres entre 50 et 100 en allant de trois en trois.

**Exercice n°7 :** Ecrire un programme qui affiche tous les carrés nombres entre -10 et 10 en allant de un en un.

**Exercice n°8 :**

Demandez à l'utilisateur la note qu'il a eu, puis si celle-ci est supérieure ou égale à 16 vous afficherez que ça correspond à un A, sinon si celle-ci est supérieure à 12 sa note correspond à un B ainsi de suite.

**Exercice n°9 :** Demander à une personne son année de naissance, et suivant celle-ci vous afficherez « tu es majeur », « tu n'es pas majeur » ou vous demanderez le numéro du mois de naissance, et de là de nouveau vous afficherez si elle est majeure ou pas ou vous demanderez le jour et vous donnerez votre verdict final.

**Exercice n°10 :** on a envie de compléter le programme commencé dans la tâche 3, on a envie que le programme nous donne au maximum 6 chances pour trouver notre nombre.

```
from random import *
cache = randint(1,100)
perdu = True
for i in range(.....):
    proposition = int(input("votre proposition ?"))
    if proposition == cache :
        print("gagné en ",i+1,"coups")
        perdu = False
        break # permet d'interrompre la boucle
    elif ..... # à compléter avec l'aide de la tâche 3
        .....
        .....
        .....
    if perdu == True
        print("après 6 essais vous n'avez pas trouvé la bonne réponse : ",cache)
```

Il existe un autre type de boucle :

## Les boucles en while

Recopier ou copier/coller le programme ci-contre. Tester le.

**Exercice n°11 :**

Ecrire un programme qui simule le lancer d'un dé à 20 faces tant que le résultat obtenu n'est pas égal à 10. Le programme doit afficher au bout de combien d'essai il a obtenu le résultat 10

```
t=10
while t>=0 :
    print(t)
    t=t-1
print("boum !!!")
```

**Exercice n°12 :**

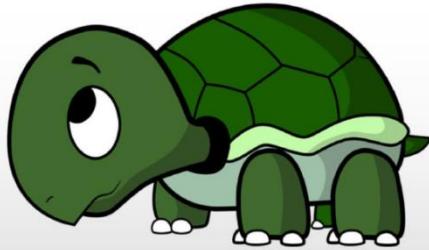
Créer un programme demandant un nombre et donnant tous les diviseurs de ce nombre.

**Exercice n°13 :**

Ecrire un programme qui demande à l'utilisateur d'entrer un nombre n et qui affiche à l'écran la table de multiplication de ce nombre. Par exemple si ce nombre est 5 on aura :  $5 \times 1 = 5$  puis à la ligne suivante :  $5 \times 2 = 10$  et ainsi de suite

# SNT – le module turtle

Le module Turtle est l'héritier du logo un très vieux langage de programmation orienté graphisme proposé aux élèves d'école primaire.



La première chose à faire est de choisir le langage Python + turtle pour votre « repl ». A la première ligne de votre programme importez la librairie Turtle :

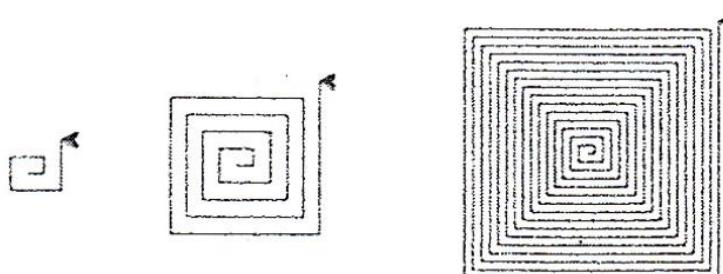
```
from turtle import *
```

Puis on utilisera les commandes suivantes :

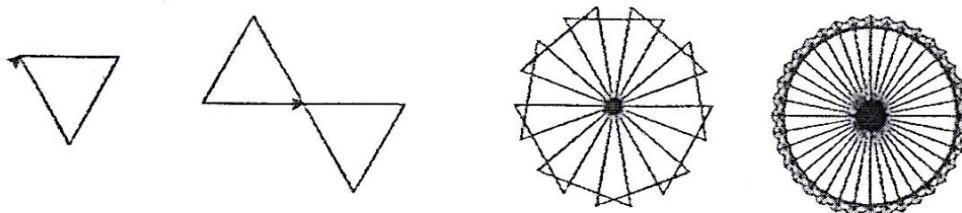
Fonction	Effet
forward(n) ou fd(n)	avance de n pixels
back(n) ou bk(n)	recule de n pixels
right(n) ou rt(n)	tourne à droite de n degrés
left(n) ou lt(n)	tourne à gauche de n degrés
clear()	efface l'écran
penup()	lève le stylo
pendown()	baisse le stylo
reset()	efface l'écran, remet la tortue au centre et réinitialise ses paramètres
showturtle()	montre la tortue
hideturtle()	cache la tortue
speed(n)	Change la vitesse de 1(lent) à 10 (rapide). La valeur spéciale 0 est la plus rapide.
update()	Force l'affichage des graphismes en attente
bye()	Referme la fenêtre
setup(w,h)	Ouvre une fenêtre de taille wxh

Voici trois listes de figures à reproduire, facile au début puis ça se complique

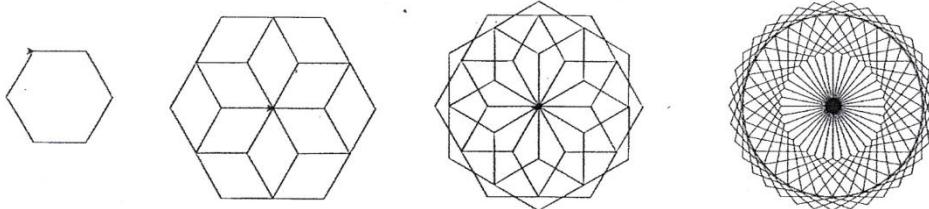
A base de segment



A base de triangles équilatéraux :



## A base d'hexagones réguliers



Faites un maximum de figure et si ça se complique trop pour vous passez à la liste suivante.  
Les figures étant répétées ça sera l'occasion pour vous de réutiliser les boucles déjà vues en classe.

Bonus : si on a envie de faire une belle animation on peut utiliser plusieurs tortues de la manière suivante :

Raphaelo = Turtle()

```
Donatello = Turtle()
Raphaelo. color(255,255,0)
Donatello.color(255,255,0)
Raphaelo.fd(50)
Donatello.bk(40)
...
```

Pour creuser un peu plus :

<https://zestedesavoir.com/tutoriels/944/a-la-decouverte-de-turtle/>

## Annexe n°1 : Premières instructions Python

Addition	+
Soustraction	-
Multiplication	*
Division	/
Quotient de la division euclidienne	//
Reste de la division euclidienne	%
Exposant	**
<b>from random import *</b>	
Nombre entier aléatoire entre a et b	randint(a,b)
Pour affecter une valeur à une variable	=

<b>from math import *</b>	
Racine carrée	sqrt(x)
Sinus	sin(x)
Cosinus	cos(x)
Tangente	tan(x)
Nombre pi	pi
PGCD des nombres a et b	gcd(a,b)
Tronquer un nombre	trunc(x)
Arrondir un nombre x à 1 chiffre après la virgule	round(x,1)
Pour tester l'égalité entre deux nombres	==