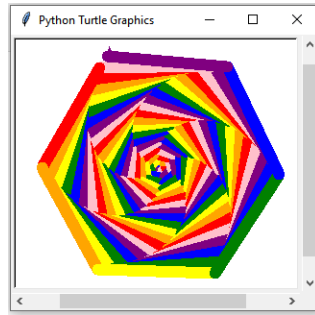


Python est un langage de programmation simple d'usage qui sera utilisé en **SNT**, en **physique-chimie**, en **mathématiques** ainsi qu'en spécialité **NSI** lors de diverses activités.

```
from turtle import *
liste = ["pink", "red",
"orange", "yellow",
"green", "blue",
"purple"]
for i in range(3*42):
    color(liste[i%7])
    pensize(10+2)
    forward(i)
    Left(59)
done()
```



Résultat du script

Utiliser Python



Python est présent sur les calculatrices récentes et plusieurs logiciels dans LORDI région : cliquer sur l'icône MCNL et rechercher **Edupython**.

Copier coller ensuite le code ci-dessus dans la fenêtre du haut, cliquer sur le triangle vert, enregistrer votre script.



Exécuter depuis une console

Dans la fenêtre du bas dans **Edupython**, saisir les commandes suivantes et essayer de comprendre leur signification

Les opérateurs mathématiques

A tester	symbole	opération
32+10	+	addition
80-38	-	
6*7	*	
2**6	**	
355/113	/	
42//11	//	
42%11	%	

Les opérateurs d'affectation

```
# à faire dans cet ordre
>>> a = 101
>>> b = 4
>>> e = a * b
>>> d = e + 2**8 + 6
>>> a -= 1
>>> b += 2
>>> a
>>> e
# puis remplir →
```

variable	valeur finale
a	
b	
d	
e	

Les opérateurs de comparaisons

A tester	symbole	comparaison réalisée
6==3**2	==	
8==2**3		
4<4	<	
4<=4	<=	
9!=9	!=	
9!=8		

Quelques fonctions natives

```
>>> len("Bonjour")
7 # Détermine la longueur
>>> max(10*5+3, 10+5*3)
#
>>> min(10*5+3, 10+5*3)
#
>>> abs(-256)
#
>>> round(1/3, 2)
#
>>> round(1/3, 5)
#
```

Les commentaires en python

Ceci est un commentaire en python

Un commentaire commence par un dièse, il n'est pas interprété lors de l'exécution du script.

Instructions conditionnelles

Une instruction conditionnelle n'est exécutée que si une condition est réalisée.

```
def ranger(a, b):
    if a < b:
        return a, b
    else:
        return b, a
```

>>> ranger(5,3)
(3, 5)

Les structures conditionnelles, les boucles, les fonctions finissent leur déclaration initiale par ":" et sont suivies de leurs instructions qui seront indentées. (décalés de 2 ou 4 espaces).

Les boucles for

L'instruction "for" est une boucle bornée, elle permet de répéter un nombre fixé de fois un bloc d'instructions indenté.

```
# Affiche la table de 7
nb = 7
for i in range(11):
    print(nb, "x", i, "=", nb * i)
```

Importer des modules en python

La fonction `sqrt` ci-dessus est la racine carrée utilisée en mathématiques, pour l'utiliser dans un script python il faut l'importer :

Pour importer toutes les fonctions d'un module :

Entraînez-vous !

>>> a = 4 >>> b = 2 >>> c = b + a >>> a = c + a >>> b = a - c	<table border="1"><thead><tr><th>variable</th><th>Valeur finale</th></tr></thead><tbody><tr><td>a</td><td></td></tr><tr><td>b</td><td></td></tr><tr><td>c</td><td></td></tr></tbody></table>	variable	Valeur finale	a		b		c	
variable	Valeur finale								
a									
b									
c									

2) Reprendre le script du sapin ci-contre à gauche puis :

b. faire un tronc aussi haut que les branches

3) Reprendre le script de la spirale

a. Faire en sorte qu'elle ressemble à un carré

b. Ajouter la couleur "cyan" et faire en sorte qu'elle s'affiche

Turtle utilise les commandes **right**(angle en degrés), **left**(angle en degrés), **penup**() et **pendown**() (pour commencer ou arrêter de tracer un trait) et **forward**(distance) pour avancer.

c. Essayer de dessiner une maison (un brouillon sur papier est fortement conseillé)

Si vous souhaitez changer la couleurs, turtle en connaît beaucoup, mais les noms sont en anglais.

Pour les afficher toutes (j'ai essayé de faire court) :

```
>>> %Run turtle01.py
```

← ce que retourne le script au-dessus

Une fonction est un sous programme que l'on peut appeler et exécuter plusieurs fois.

```
# start.py
from math import sqrt

def hypotense(a, b):
    # Théorème de Pythagore
    c = sqrt(a ** 2 + b ** 2)
    # La fonction renvoie le résultat
    return c

def est_negatif(n):
    return n <= 0
```

Recopier le résultat des commandes suivantes

```
>>> hypotenuse(3, 4)
```

```
>>> est_negatif(-5)
```

```
>>> est_negatif(3)
```

```
from matplotlib import colors
#le module matplotlib permet de lister les couleurs,
#je les récupère dans liste_couleurs
liste_couleurs=list(colors.cnames.keys())
#je regarde quel est le nom le plus long pour aligner proprement
max_long=max([len(i) for i in liste_couleurs])
#je vais les aligner sur 6 colonnes
for i in range(len(liste_couleurs)//6+1):
    for j in range(6):
        if i*6+j<len(liste_couleurs):
            print(liste_couleurs[i*6+j],
                  end=(" "*(max_long-len(liste_couleurs[i*6+j])))+ " | ")
#un retour à la ligne vide pour changer de ligne
print()
```