

Dessiner avec turtle

Christophe Viroulaud

Première - NSI

Lang 04

Pour faciliter le travail des développeurs Python, il existe des outils spécialisés dans diverses tâches. On les appelle *bibliothèques*, *modules* ou encore *librairies*. Comme son nom l'indique la bibliothèque `math` offre des fonctionnalités permettant d'effectuer des calculs mathématiques.

Comment utiliser une bibliothèque ?

La documentation

Importation

Une bibliothèque
graphique

Découverte

Premiers déplacements

Des figures plus complexes

1. La documentation

2. Importation

3. Une bibliothèque graphique

Il n'est pas nécessaire de connaître par cœur toutes les possibilités de chaque librairie. Il est par contre indispensable de savoir utiliser la documentation en ligne de Python.

Activité 1 :

1. Se rendre sur la page
<https://docs.python.org/3/>
2. Sélectionner la langue et la version de Python correspondante à l'EDI utilisé.
3. Dans la barre de recherche, taper `math` et ouvrir le premier lien.
4. Chercher la fonction permettant de calculer la racine carrée d'un nombre.

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

```
1 from math import sqrt
2 print(sqrt(25))
```

Code 1 – Racine carré

La documentation

Importation

Une bibliothèque
graphique

Découverte

Premiers déplacements

Des figures plus complexes

1. La documentation

2. Importation

3. Une bibliothèque graphique

Pour utiliser les fonctionnalités proposées par une bibliothèque, il faut d'abord l'importer dans le programme. Plusieurs possibilités d'import existent.

```
1 # importe toute la bibliothèque
2 import math
3 # calcule le cosinus de l'angle (en radians)
4 c = math.cos(0.5)
```

```
1 # importe toute la bibliothèque et lui donne un
  alias
2 import math as m
3 c = m.cos(0.5)
```

```
1 # importe toutes les fonctions de la bibliothèque
2 from math import *
3 # Il ne faut plus faire référence au nom de la
  bibliothèque
4 c = cos(0.5)
```

```
1 # n'importe que les fonctions nécessaires dans le
  programme
2 from math import cos
3 # Il ne faut plus faire référence au nom de la
  bibliothèque
4 c = cos(0.5)
```

Activité 2 :

1. Écrire un programme qui renvoie le cosinus, le sinus et la valeur en degrés d'un angle $\frac{\pi}{2}$.
2. Comment expliquer la valeur du cosinus ?

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

```
1 from math import cos, sin, degrees, pi
2
3 angle = pi/2
4 print(f"en degré: {degrees(angle)}, \ncos: {
    cos(angle)},\nsin: {sin(angle)}")
```

Code 2 – Trigonométrie

Mathématiquement $\cos \frac{\pi}{2} = 0$. La représentation des nombres réels en mémoire peut poser problème.

La documentation

Importation

Une bibliothèque
graphique

Découverte

Premiers déplacements

Des figures plus complexes

1. La documentation

2. Importation

3. Une bibliothèque graphique

3.1 Découverte

3.2 Premiers déplacements

3.3 Des figures plus complexes

La bibliothèque `turtle` est un module simple pour réaliser des figures géométriques. La *tortue* avance, tourne sur l'écran et trace les traits demandés par l'utilisateur.

La documentation

Importation

Une bibliothèque
graphique

Découverte

Premiers déplacements

Des figures plus complexes

1. La documentation

2. Importation

3. Une bibliothèque graphique

3.1 Découverte

3.2 Premiers déplacements

3.3 Des figures plus complexes

Les possibilités sont nombreuses. Il faut d'abord découvrir quelques déplacements.

Activité 3 :

1. Dans la documentation, chercher le rôle des fonctions :
 - ▶ `forward`, `backward`
 - ▶ `left`, `right`
 - ▶ `up`, `down`
2. Tracer un carré de 100 de côté.

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

```
1 import turtle as t
2
3 for _ in range(4):
4     t.forward(100)
5     t.left(90)
6 t.done()
```

Code 3 – Carré

La documentation

Importation

Une bibliothèque
graphique

Découverte

Premiers déplacements

Des figures plus complexes

1. La documentation

2. Importation

3. Une bibliothèque graphique

3.1 Découverte

3.2 Premiers déplacements

3.3 Des figures plus complexes

Activité 4 :

1. Réaliser la figure 1.
2. Remplir chaque carré avec une couleur, de la manière suivante :
 - rouge si le numéro du carré tracé est impair,
 - vert s'il est pair.

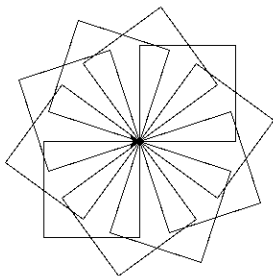


FIGURE 1 – Des carrés et des rotations

Avant de regarder la correction



- ▶ Prendre le temps de réfléchir,
- ▶ Analyser les messages d'erreur,
- ▶ Demander au professeur.

```
1 import turtle as t
2
3 for i in range(10):
4     # on trace le carré
5     for j in range(4):
6         t.forward(100)
7         t.left(90)
8
9     # on tourne de 36° pour faire le tour complet
10    t.left(36)
11
12 t.done()
```

Code 4 – Rosaces


```
1  for i in range(10):
2      # i est pair (le reste de la division par 2
   est nul)
3      if i%2 == 0:
4          t.color("green", "green")
5      else:
6          t.color("red", "red")
7
8      t.begin_fill()
9      for j in range(4):      # on trace le carré
10         t.forward(100)
11         t.left(90)
12
13     t.end_fill()
14     t.left(36)
15 t.done()
```

Code complet

Dessiner avec
turtle

La documentation

Importation

Une bibliothèque
graphique

Découverte

Premiers déplacements

Des figures plus complexes

Le code complet est accessible [ici](#).