

Module HAT004
Projet calcul / Projet personnel avance
Accompagnement python: Les bases du langage
Python - Orienté Objet
Session 02

Ronan Dupont

1 Création d'une classe point

Pour cet exercice, on importera les packages `numpy`, `matplotlib` à l'aide de la commande:

```
from matplotlib.pyplot import *
```

Question 1 – Créer une classe `Point` qu'on initialisera avec les coordonnées x et y . Vous devriez pouvoir initialiser le point p_0 avec `p0 = Point(0,0)`. Initialisez également le point `p1 = Point(0,2)`.

Question 2 – Dans cette classe, créer une fonction `aff_pt(self)` qui affichera le texte suivant:

```
p0.aff_pt()  
>> x= 0 ; y= 0
```

Question 3 – Créer une fonction `dist(self,pt)` renvoyant la distance entre deux points. Par exemple:

```
print(p0.dist(p1))  
>> 2
```

2 Création d'une classe rectangle

Question 4 – Initialisez maintenant les points `p2 = Point(4,2)` et `p3 = Point(4,0)`.

Question 5 – Créer une classe `Rectangle` prenant 4 points en initialisation. Vous pourrez l'initialiser via la commande: `R1 = Rectangle(p0,p1,p2,p3)`.

Question 6 – En réutilisant la question 3, créer une fonction `air` qui va renvoyer l'air du rectangle. Quelle est l'air du rectangle ?

Question 7 – En réutilisant la question 3, créer une fonction `périmètre` qui va renvoyer le périmètre du rectangle. Quel est le périmètre du rectangle ?

Question 8 – En utilisant la fonction `scatter(x0,y0)`, créer une fonction `plot_pts(self)` qui va afficher tous les points du rectangle.

Question 9 – En utilisant la fonction `plot(X,Y)`, créer une fonction `plot_rect(self)` qui va afficher le rectangle.

3 Création d'une classe de nuage de points

Question 10 – Créer une classe `Nuage` qui prend en argument une liste L qui va contenir des points.

Question 11 – Créer les points suivants:

Point	p1	p2	p3	p4	p5	p6	p7	p8
x=	0	0	2	4	4	10	10	8
y=	0	4	2.5	4	0	0	4	2.5

et initialiser une liste L1 avec les 5 premiers points puis une seconde liste L2 avec les 3 derniers.

Question 12 – Initialiser deux objets nuages avec la commande: `N1 = nuage(L1)` et de même pour N2.

Question 13 – Créer une fonction `plot_pts(self)` qui va afficher les points. Tester la commande avec `N1.plot_pts()`

Question 14 – Créer une fonction `contour_pts(self)` qui va tracer les points reliés entre eux.

Question 15 – Exécuter la commande les commandes:

```
N1.contour_pts()
N2.contour_pts()
show()
```

et observer la forme à l'écran. Que pouvez-vous lire ?

Question 16 – Bonus: Créer une fonction `distance` dans la classe `nuage` puis trouver la distance totale pour l'affichage ci-dessus.

Pour plus d'exercices, voir par exemple:

<https://docplayer.fr/194014571-Tp-programmation-orientee-objet-en-python.html>