

Les fonctions en python

Cours

Les fonctions permettent de mettre à part un morceau de code en lui donnant un nom, pour pouvoir le réutiliser juste en appelant ce nom.

1) Fonctions sans paramètres

Si on veut souvent dessiner un triangle, on peut créer la fonction "triangle" comme ceci :

```
def triangle():  
    for i in range(3):  
        forward(100)  
        right(120)
```

Puis à chaque fois qu'on veut tracer un triangle on écrira juste : triangle()

Attention comme pour **if**, **else**, **while**, **for**, avec **def** il faut:

- mettre deux points ":" à la fin de la ligne
- décaler toutes les lignes suivantes qui font partie de la fonction à droite avec la touche TAB

2) Fonctions avec paramètre(s)

On peut nommer un paramètre entre les parenthèses de la fonction, puis l'utiliser dans la fonction comme une variable.

Par exemple notre fonction triangle dessinait toujours des carrés de côté 100. Si on veut pouvoir choisir la longueur du triangle quand on l'utilise, on peut la changer comme ça :

```
def triangle(longueur):  
    for i in range(3):  
        forward(longueur)  
        left(120)
```

Puis si on veut dessiner un triangle de taille 40 on écrira : triangle(40)

Il est possible d'avoir plusieurs paramètres séparés par des virgules.

3) Fonctions avec valeur de retour

Une fonction peut renvoyer une valeur avec le mot-clé **return**

Par exemple la fonction suivante calcule la surface d'un disque :

```
def disque(rayon):  
    return 3.14 * rayon * rayon
```

Si on l'utilise en écrivant "disque(12)" par exemple, la fonction est appelée et disque(12) est remplacé par le résultat.

Si on veut afficher le résultat on utilisera : print(disque(12))

Exercices

Exercice 1

1) Lire la partie 1 du cours.

2.a) Sur un navigateur allez à l'adresse <https://concours.castor-informatique.fr/?team=SNTturtp> et allez à la **version 1 étoile** de l'exercice 4 "Procédures".

b) Définissez une fonction `carre()` qui dessine un carré, et appelez-la pour dessiner le premier carré.

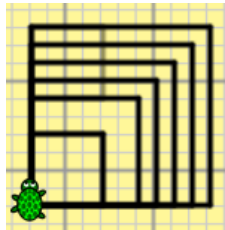
c) Appelez en tout quatre fois la fonction en avançant entre chaque appel.

d) Pour que la tortue ne dessine pas entre les carrés, rajouter `lowerBrush()` sur une ligne au début de la fonction (après la ligne du `def`), et `liftBrush()` à la fin de la fonction. Normalement l'exercice est validé !

Exercice 2

1) Lire la partie 2 du cours.

2) Revenez à la version 1 étoile de l'exercice "Procédures" que vous venez de valider, changez votre fonction `carre()` pour qu'elle prenne la taille du carré en paramètre, et utilisez-la pour dessiner le dessin suivant :



Exercice 3

1) Lire la partie 3 du cours.

2.a) Fermez le navigateur et ouvrez le logiciel edupython.

b) Recopiez la fonction `disque` du cours et utilisez la pour afficher la surface d'un disque de rayon 7.

Exercice 4

1) Recopiez ce code dans edupython :

```
from turtle import *

def flocon(niveau, longueur):
    if niveau == 0:
        forward(longueur)
    else
        pass
```

Puis appelez la fonction `flocon(0,100)`, et ajoutez sur une dernière ligne `exitonclick()`

Que se passe-t-il ? (grâce à la dernière ligne la fenêtre graphique se ferme si vous cliquez dessus)

2) On veut que si le paramètre `niveau` vaut 1, ce dessin soit tracé :



Pour cela, recopiez ce code à la place de "pass" dans la fonction précédente :

```
forward(longueur/3)
left(60)
forward(longueur/3)
right(120)
forward(longueur/3)
left(60)
forward(longueur/3)
```

Vérifiez que ça fonctionne en appelant `flocon(1,100)`

3) Remplacez chaque "`forward(longueur/3)`" dans le code par "`flocon(niveau-1,longueur/3)`"

Que se passe-t-il si vous appelez `flocon(1,100)` ? `flocon(2,100)` ? `flocon(3,100)` ?...