

## Python NSI sierpinski

```
import turtle
turtle.tracer(0,0)
turtle.screensize(2000,2000)
turtle.pu()
turtle.goto(-500,0)
turtle.pd()

def dessiner(courbe, longueur, angle):
    for caractere in courbe:
        if caractere == '+': turtle.left(angle)
        elif caractere == '-': turtle.right(angle)
        elif caractere in ['F', 'G']: turtle.forward(longueur)

def regleSierpinski(chaine):
    nouvelleChaine = ""
    for lettre in chaine:
        if lettre == 'F':
            nouvelleChaine = nouvelleChaine + 'F-G+F+G-F' ## si on a F il devient F-G+F+G-F
        elif lettre == "G":
            nouvelleChaine = nouvelleChaine + "GG" ##si on a un G il devient GG
        else :
            nouvelleChaine = nouvelleChaine + lettre
    return nouveleChaine

def courbeSierpinski(motifInitial, niter): ##changer nom
    courbe = motifInitial
    for i in range(niter):
        nouveauMotif = regleSierpinski(courbe) ##changer nom
        courbe = nouveauMotif
    return courbe

def flocon(motifInitial, niter):
    courbe = courbeSierpinski (motifInitial, niter)
    flocon = ""
    for _ in range(3):
        flocon += courbe
        flocon += '--'
    return flocon

longueur = 5
angle = 120
niter = 6
dessiner(courbeSierpinski('F-G-G', niter), longueur, angle) ##car F-G-G = base

turtle.update()
```

```
turtle.exitonclick()
```