Répéter une instruction

Objectif :

- répéter des instructions avec la boucle TANT QUE (ou WHILE) ou avec la boucle POUR (ou FOR).
- organiser son algorithme à l'aide d'une fonction/procédure.

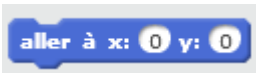
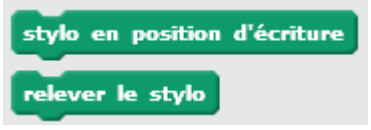
Principe :
Utiliser un dessin avec le module turtle – De ce fait la toute première ligne de votre programme sera

```
1 from turtle import *
```

Pour éviter les bugs la dernière ligne de votre programme sera :

```
exitonclick()
```

Dans ce module « turtle », on retrouve les instructions suivantes

Instruction (en italique, les valeurs sont modifiables).	**Instruction du module turtle** (en italique, les valeurs sont modifiables).	**équivalent scratch**
Réinitialiser tout Effacer tout	reset() clear()	
Avancer de 10 Reculer de 10	forward(*10*) backward(*10*)	
tourner à droite de 90° tourner à gauche de 90°	right(*90*) left(*90*)	
s'orienter à 90°	setheading(*90*)	
se positionner au point (0;0)	goto(0,0)	
abaisser le « stylo » relever le « stylo »	down() ou pendown() up() ou penup()	
couleur du stylo en gris (rouge – bleue -jaune etc...)	color(« grey») (« red », « blue », « yellow » etc..)	
épaisseur du stylo à 1	width(1)	

Dessiner un carré de 20 pixels en utilisant une boucle.

Et plus d'instruction encore en tapant dans la console :

```
import turtle

help(turtle)
```

Répéter des carrés

Voici deux fonctions (ou procédure) informatique qui trace un carré. Quelle est la différence ?

```
def carre(c):
    down()
    k=1
    setheading(0)
    while k<=4:
        forward(c)
        right(90)
        k=k+1
    up()
    return
```

```
def carre(c):
    down()
    setheading(0)
    for i in range (4) :
        forward(c)
        right(90)
    up()
    return
```

Remarque :

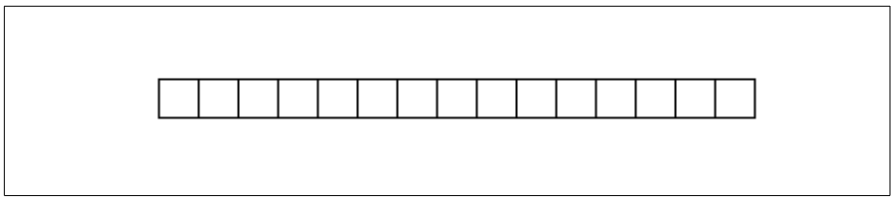
- le fait d'utiliser une fonction/procédure permet d'intégrer des variables locales (ici la longueur de mon carré va être appelée à 20 pixels et automatiquement mis dans la variable locale « c »))
- cela permet de commencer à « ranger » mon algorithme lorsque je vais devoir appeler la fonction/procédure « carre ».

Exemple d'utilisation : carre_aux_coins.py (dans votre espace classe sur le réseau)

Exercice 1 : la ligne de carrés.

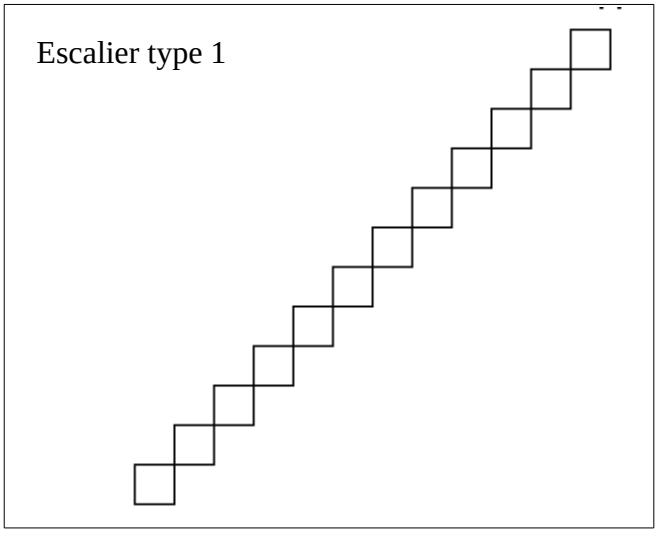
En appelant la fonction-procédure « carre(20) » dans votre programme principal, dessiner les figures suivantes constituer de 15 carrés côte à côte

On pourra commencer à goto(-50,0)

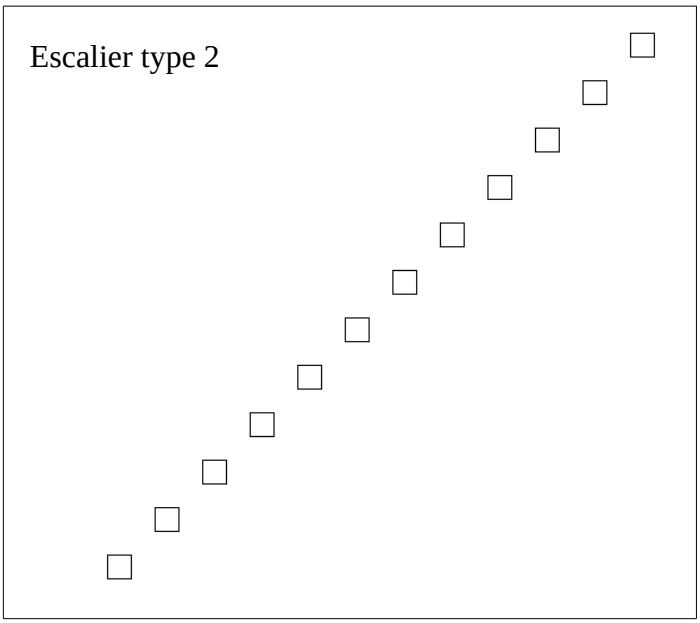


Exercice 2 : les escaliers de carrés.

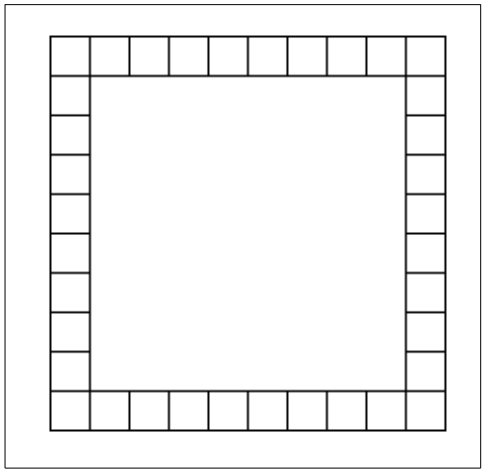
Escalier type 1



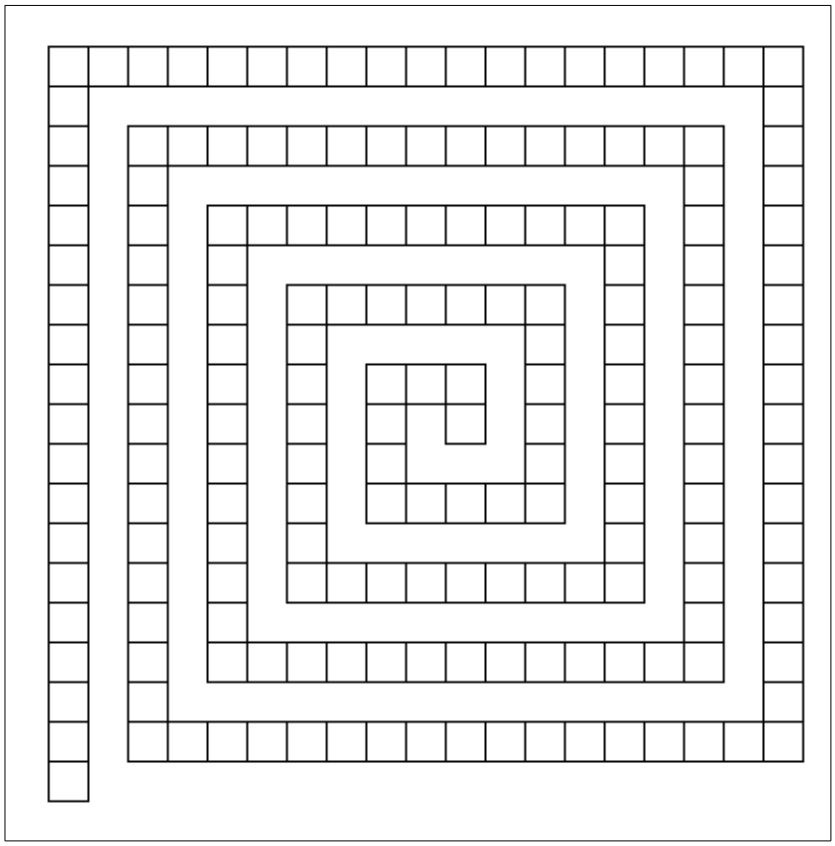
Escalier type 2



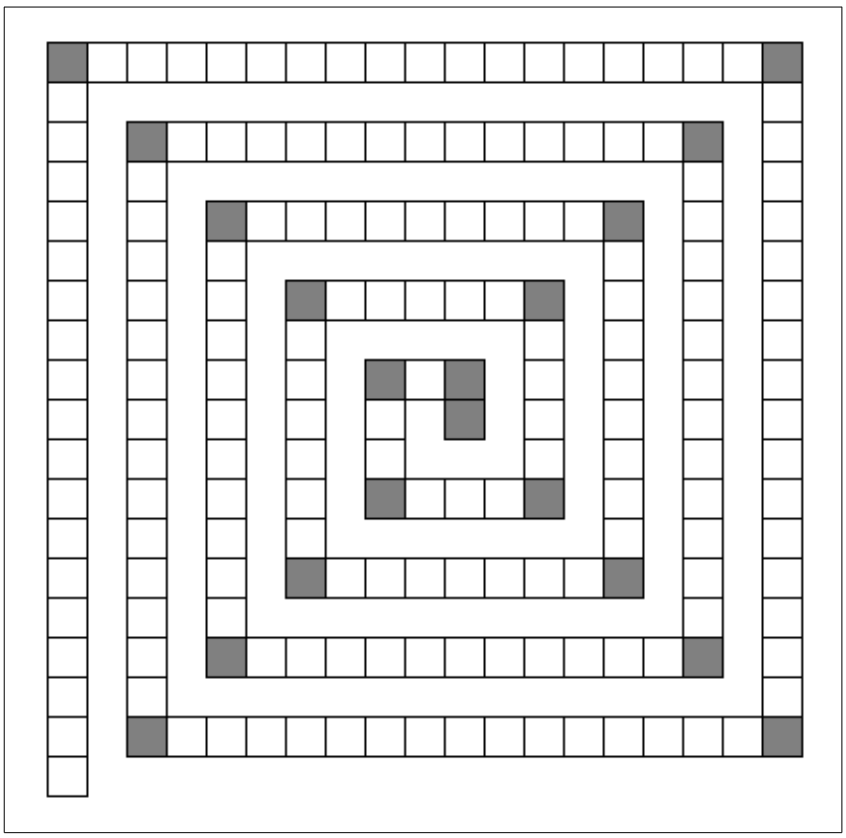
Exercice 3 : Un carré de carrés



Exercice 4 : la spirale de carrés



Exercice 5 : la spirale de carrés colorisés aux coins.

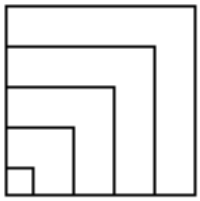


Pour colorier l'intérieur d'un carré de côté 20

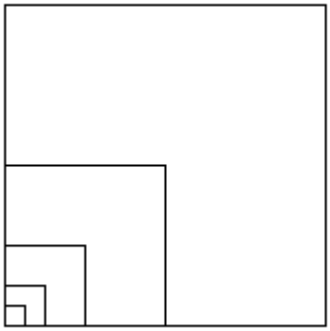
```
color("black","grey")
begin_fill()
carre(20)
end_fill()
```

Exercice 6 : les carrés imbriqués.

cas 1 : on augment la longueur de façon régulière :



cas 2 : on double la longueur à chaque itération.



Exercice 7 : les carrés empilés – non testé avec les élèves.

Le choix du nombre de carrés de la première et de la dernière ligne peut être demandé, ainsi que la longueur d'un côté.