

Répétitions (activité débranchée)

Le code python de cette activité utilise plusieurs fonctions de la bibliothèque *turtle* :

- *forward(p)* qui avance le curseur de *p* pixels. Au démarrage du programme (si aucune instruction ne la précède) on avance vers la droite.
- *left(a)* qui oriente le curseur de *a* degrés dans le sens inverse des aiguilles d'une montre.
- *right(a)* qui oriente le curseur de *a* degrés dans le sens des aiguilles d'une montre
- *up()* qui permet de lever le crayon et *down()* qui abaisse le crayon

Répétition – Boucle POUR

Les scripts scratch et python suivants permettent d'obtenir le même dessin

SCRATCH



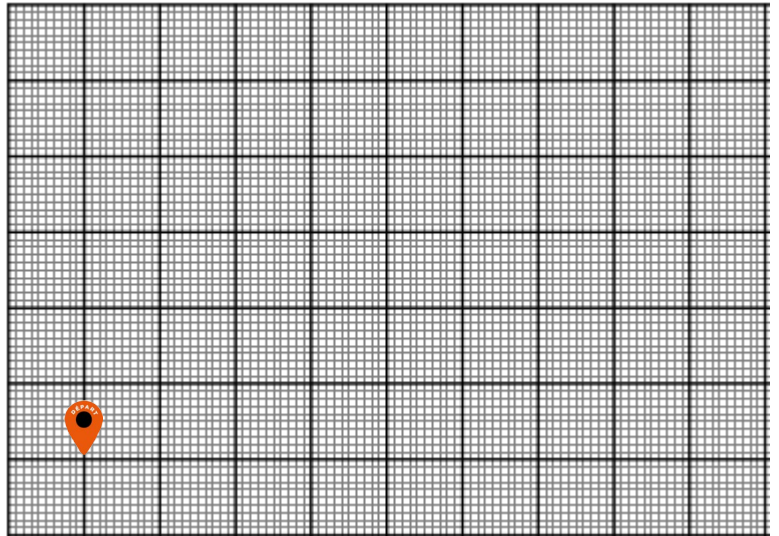
```
1 from turtle import *
2
3 for compteur in range(4):
4     forward(50)
5     right(90)
```

Pour chaque script suivant, esquissez le dessin obtenu après exécution du script.

On choisit comme échelle : **1pixel est équivalent à 1 mm**

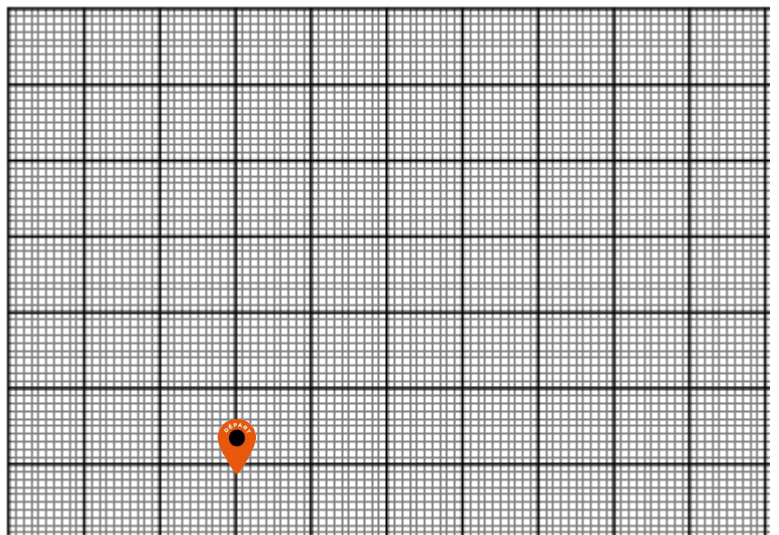
Script 1 :

```
1 from turtle import *
2
3 longueur = 50
4
5 for _ in range(10):
6     forward(longueur)
7     left(90)
8     longueur = longueur - 5
```



Script 2 :

```
1 from turtle import *
2
3 longueur = 30
4
5 for _ in range(5):
6     forward(longueur)
7     left(72)
```

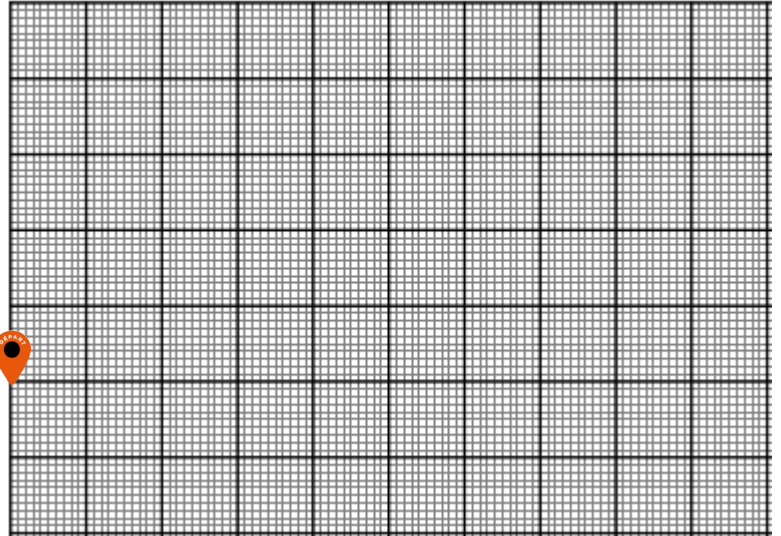


Répétitions (activité débranchée)

Script 3 :

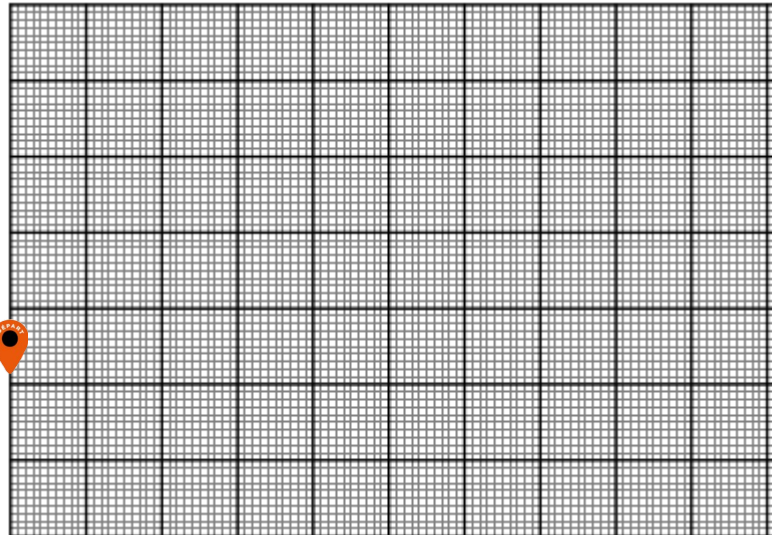
On considère la fonction `carre` qui prend comme paramètre `longueur`

```
1 from turtle import *
2
3 def carre(longueur):
4     for compteur in range(4):
5         forward(longueur)
6         left(90)
7
8 for _ in range(10):
9     carre(10)
10    up()
11    forward(15)
12    down()
```



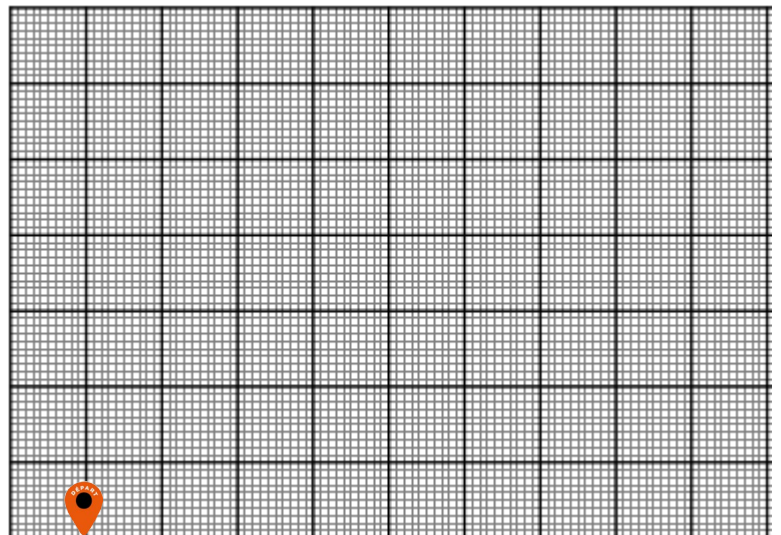
Script 4 :

```
1 from turtle import *
2
3 def carre(longueur):
4     for _ in range(4):
5         forward(longueur)
6         left(90)
7
8 longueur = 5
9 for _ in range(10):
10    carre(longueur)
11    up()
12    forward(longueur + 5)
13    down()
14    longueur = longueur + 5
```



Script 5 :

```
1 from turtle import *
2
3 def carre(longueur):
4     for compteur in range(4):
5         forward(longueur)
6         left(90)
7
8 longueur = 5
9 for _ in range(10):
10    carre(longueur)
11    longueur = longueur + 5
```



Répétitions (activité débranchée)

Répétition sous conditions – Boucle TANT QUE

SCRATCH :



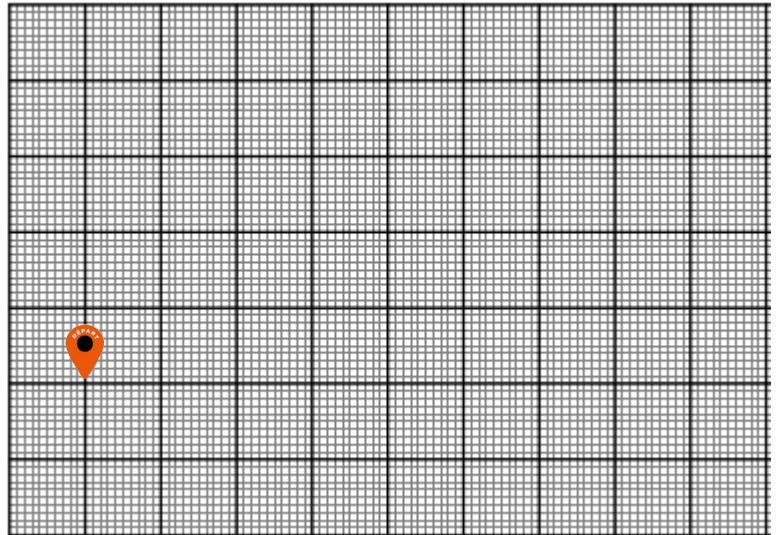
Python

```
1 from turtle import *
2
3 longueur = 100
4
5 while longueur >= 5:
6     forward(longueur)
7     right(90)
8     longueur = longueur - 10
```

Pour chaque script suivant, dessiner sur votre feuille le dessin obtenu après exécution.
On choisit comme échelle : 1pixel est équivalent à 1 mm (ce qui n'est pas vrai sur un écran)

Script 6 :

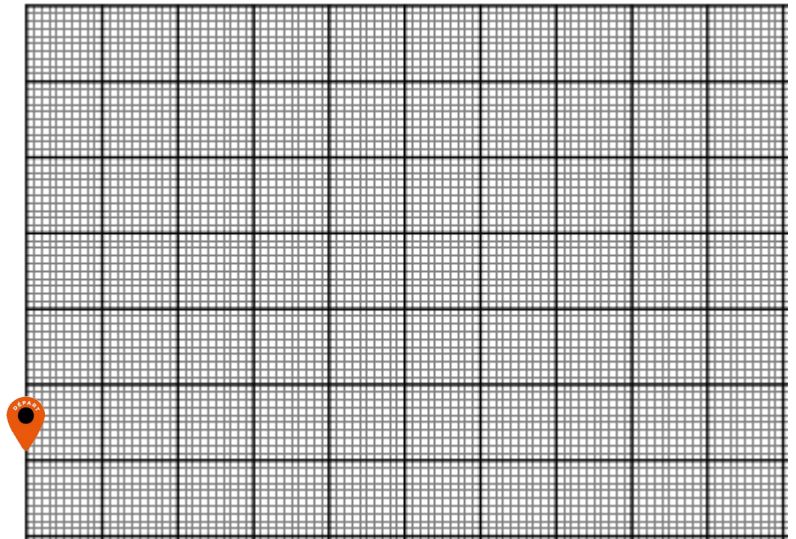
```
1 from turtle import *
2
3 def carre(longueur) :
4     for _ in range(4):
5         forward(longueur)
6         left(90)
7
8 cote = 10
9 nombreFrise = 5
10
11 while nombreFrise >= 0:
12     carre(cote)
13     nombreFrise = nombreFrise - 1
14     up()
15     forward(15)
16     down()
```



Répétitions (activité débranchée)

Script 7:

```
1 from turtle import *
2
3 def carre(longueur) :
4     for _ in range(4):
5         forward(longueur)
6         left(90)
7
8 cote = 50
9
10 while cote >= 0:
11     carre(cote)
12     up()
13     forward(cote + 5)
14     down()
15     cote = cote - 15
```



Les deux scripts suivants donnent-ils le même résultats ? :

```
1 from turtle import *
2
3 marche = 10
4
5 for _ in range(7):
6     forward(marche)
7     right(90)
8     forward(marche)
9     left(90)
```

```
1 from turtle import *
2
3 marche = 10
4 compteur = 0
5
6 while compteur < 7:
7     forward(marche)
8     right(90)
9     forward(marche)
10    left(90)
11    compteur = compteur + 1
```

Les deux scripts suivants donnent-ils le même résultats ? :

```
1 from turtle import *
2
3 cote = 50
4
5 for _ in range(7):
6     forward(cote)
7     right(120)
8     cote = cote - 5
```

```
1 from turtle import *
2
3 cote = 50
4
5 while cote >= 15:
6     forward(cote)
7     right(120)
8     cote = cote - 5
```