



Le module Turtle permet d'avoir un environnement simple muni d'une interface graphique.

Exercices de niveau facile et intermédiaire

### Premier programme: dessiner un carré

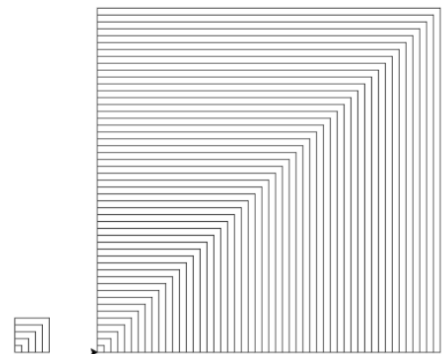
TI 83 CE Premium Python	Android - Pydroid3
<pre> 1. from turtle import * 2. t = Turtle() 3. t.clear() 4. for i in range(4): 5.     t.forward(100) 6.     t.left(90) 7. t.done()                     </pre>	<pre> 1. from turtle import * 2. t = Turtle() 3. t.clear() 4. t.shape("turtle") 5. for i in range(4): 6.     t.forward(100) 7.     t.left(90) 8. exitonclick()                     </pre>

### Utiliser une procédure

Proposez une procédure **SQUARE** qui dessine un carré dont la longueur des côtés est fournie en paramètre. Ce carré sera dessiné depuis l'état courant de la tortue. À la fin du dessin l'état de la tortue sera identique à son état initial.

### Dessiner plusieurs carrés

Proposez une procédure **cinq\_carres()** qui dessine cinq carrés emboîtés ayant comme sommet commun la position initiale de la tortue. Les longueurs des carrés sont comprises entre 10 à 50 (inclus). Les longueurs de chacun des carrés varient de 10 en 10.

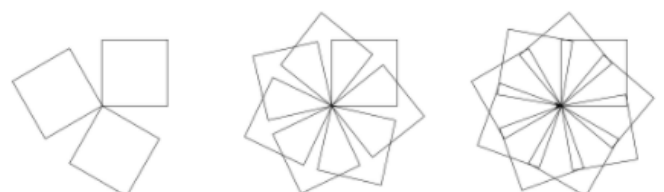


Vous pouvez également proposer une variante **vingt\_carres()** qui dessine cinquante carrés de longueurs comprises entre 10 et 200 inclus. C'est plus joli, mais il est plus difficile de vérifier qu'il y a bien 20 carrés dessinés.

### Carrés tournants

Proposez une procédure **carres\_tournants()** qui dessine n carrés de côté 100 pivotant autour d'une sommet commun pour faire un tour complet.

L'angle entre deux carrés successifs est donc de  $360 \div n$  degrés. Cette procédure est paramétrée par le nombre de carrés à dessiner. Voici le résultat pour 3, 7, et 9 carrés.

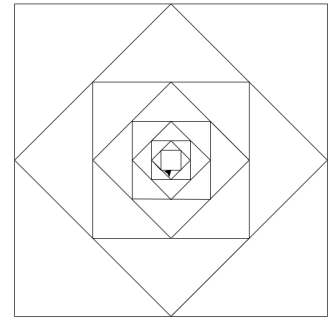




## Carrés emboîtés

Proposez une procédure `carres_emboites()` qui dessine de manière répétée un certain nombre de carrés emboîtés comme dans la figure ci-dessous (exemple pour 9 carrés).

Cette procédure admet un paramètre, la longueur du carré le plus grand. Le dessin peut se faire en dessinant les carrés depuis le plus grand jusqu'au plus petit. La taille de deux carrés successifs est alors dans un rapport de  $\sqrt{2}$ .



## Dessiner avec un ordre

On souhaite donner une *séquence* d'ordres à la tortue sous forme d'une chaîne de caractères. Étant donné

- une longueur  $l$
- un angle  $\alpha$  exprimé en degrés

La tortue interprètera chacun des caractères de la chaîne comme une commande à effectuer :

caractère	ordre
F	avance d'une longueur l
+	tourne vers la gauche de l'angle défini $\alpha$
-	tourne vers la droite de l'angle défini $\alpha$

Par exemple, la chaîne de caractères `'++F-F--F-F'` pour des valeurs de  $l=100$  et  $\alpha=45^\circ$  permettrait de tracer une « maison ».

Proposez une procédure `dessine()` qui prend en paramètres :

- une chaîne de caractères, séquence d'ordres (un mot formé d'une suite de caractères F, +, -)
- une longueur,
- un angle,

...et réalise à l'aide la tortue le tracé correspondant aux ordres.

Testez cette procédure sur les exemples précédents (dessin d'un carré, de vingt carrés, et carrés emboîtés).

## Exercices de niveau avancé

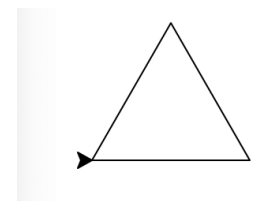
### Dessiner des polygones

Proposez une procédure pour dessiner un polygone régulier. Cette procédure acceptera un paramètre entier indiquant le nombre de côtés  $n$ , et un paramètre correspondant à la longueur  $l$  de la figure à tracer.

Par exemple

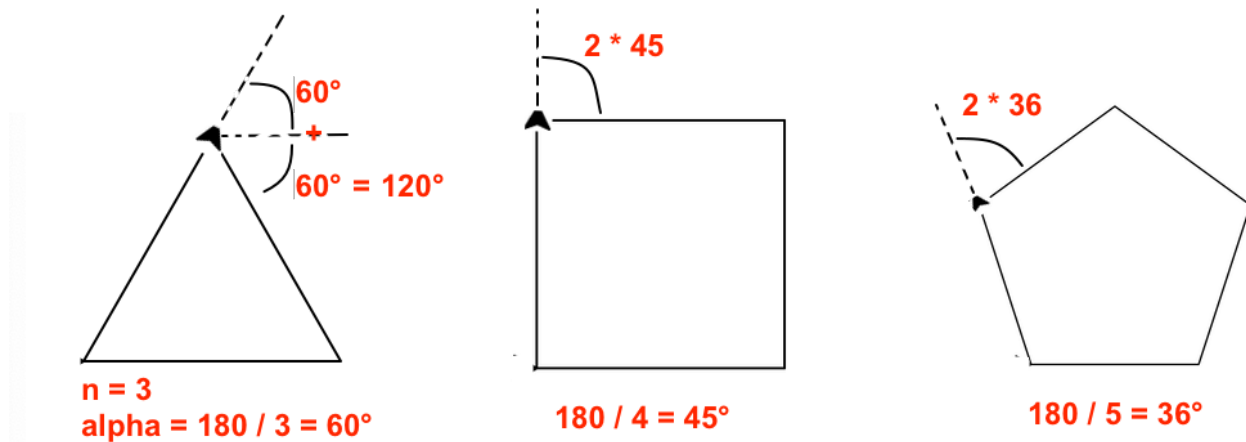
```
>>> dessine_un_polygone(3,100):
```

...produira le dessin suivant:





On déduira du schéma suivant une formule de récurrence entre l'angle de rotation lors du tracé et le nombre de côtés n:



## Dessiner avec un ordre des polygones

1. Proposez une procédure utilisant celle précédente, `dessine()`, pour dessiner un polygone régulier de côté n avec un ordre.

Cette procédure acceptera un paramètre entier indiquant le nombre de côtés, et un paramètre correspondant à la longueur de la figure à tracer.

2. Proposez une procédure qui trace l'ensemble des polygones réguliers entre deux nombres de côtés n et m donnés en paramètre.

Par exemple

```
>>> dessine_des_polygones(3, 12, 100):
```

...produira le dessin suivant:

