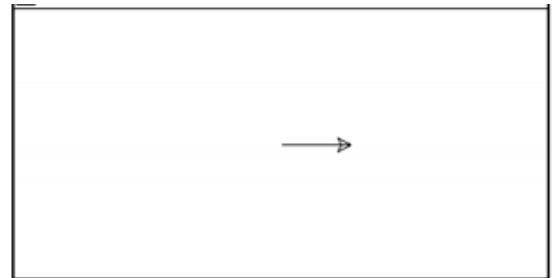


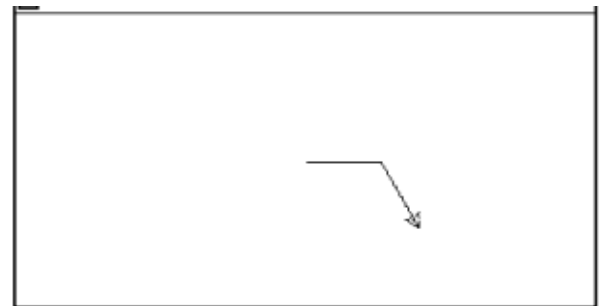
# from turtle import\*

**FORWARD** : Dans la bibliothèque turtle, il faut considérer que l'on guide un crayon symbolisé par une flèche ou une tortue. Ce crayon est par défaut en position d'écriture. La fonction forward permet de tracer un trait d'un certain nombre de pixels dans la direction de la flèche (vers la droite par défaut).

Ainsi, forward(50) trace un trait de 50 pixels.

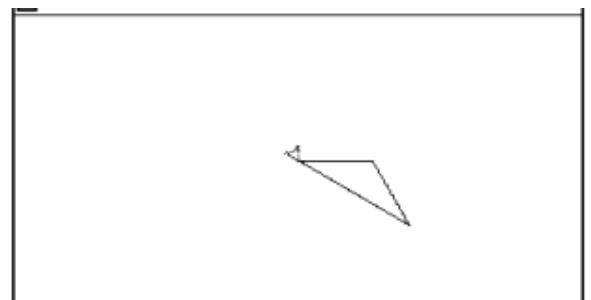


**RIGHT / LEFT** : Les fonctions right et left permettent de modifier la direction d'un nombre donné de degrés dans le sens trigonométrique (pour left) ou des aiguilles d'une montre (pour right). Dans l'exemple ci-contre, nous avons successivement tourné de 60° dans le sens des aiguilles d'une montre **right(60)** puis avancé une nouvelle fois de 50 pixels **forward(50)**. La direction n'est plus la même.



**GOTO** : Pour aller directement à un point de coordonnées (x ; y), il faut utiliser la fonction goto. Celle-ci prend 2 arguments : l'abscisse x et l'ordonnée y du point d'arrivée. Puisque le crayon est encore baissé, nous allons tracer un segment jusqu'à notre point d'arrivée (ici le point de coordonnées (0 ; 0)). La direction de la flèche n'a ici aucune influence. Les coordonnées x et y sont en nombre de pixels.

```
forward(50)
right(60)
forward(50)
goto(0,0)
```



```
· forward(50)
· right(60)
· forward(50)
· goto(0,0)
· back(50)
· up()
· back(30)
```

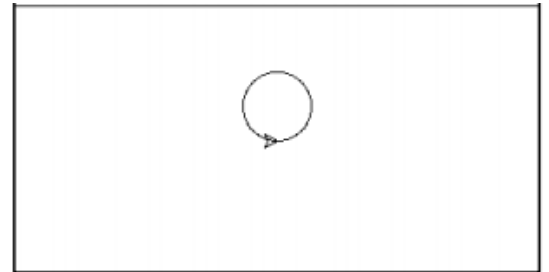
**BACK** : La fonction back permet de reculer d'un nombre donné de pixels. Exemple : `back(50)`

**UP** : La fonction up permet de lever le crayon. Ainsi, il est possible de déplacer le curseur sans tracer quoi que ce soit. Cette fonction ne prend pas d'argument. Exemple : `up()` suivi de `back(30)` : le curseur recule sans tracer de segment.

**CIRCLE** : La fonction circle permet de tracer un cercle (ou un bout de cercle) de rayon donné. La fonction `circle(25)` tracera un cercle de rayon 25 pixels et dont l'inclinaison du cercle est donnée par l'angle de départ.

**SHAPE** : Il est possible de modifier la forme du curseur avec la fonction shape. Cette fonction prend un seul argument. Deux possibilités s'offrent à nous, la forme "classic" (en forme de chevron, forme par défaut) ou la forme "turtle" (en forme de tortue).

**hideturtle()**  
masque la tortue.



```
· shape("turtle")
```

2.

```

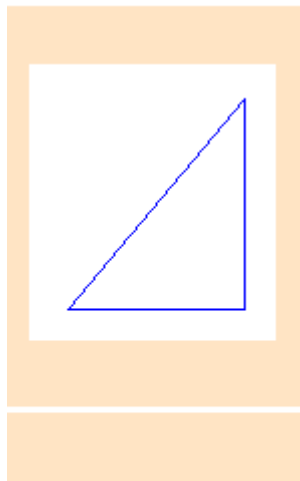
quand  pressé
répéter 1 fois
  abaisser le stylo
  avancer de 30 pas
  relever le stylo
  avancer de 10 pas
  
```

Quelle valeur faut-il saisir dans ce script pour obtenir la figure ci dessous ?

\_\_\_\_\_

```

quand  pressé
  abaisser le stylo
  avancer de 88 pas
  tourner de 90 degrés
  avancer de 105 pas
  tourner de 140 degrés
  avancer de 137 pas
  
```



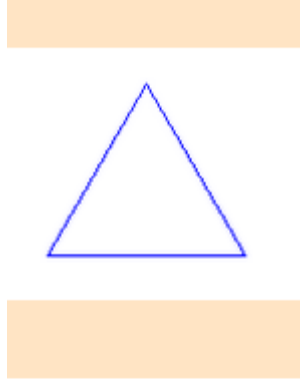
```

from turtle import*
forward(88)
left(90)
forward(105)
left(140)
forward(137)
hideturtle()
  
```



```

quand  pressé
  abaisser le stylo
  répéter 3 fois
    avancer de 100 pas
    tourner de 120 degrés
  
```



```

from turtle import*
for k in range(3):
  forward(100)
  left(120)
hideturtle()
  
```

```

quand  pressé
  abaisser le stylo
  répéter 2 fois
    avancer de 100 pas
    tourner de 90 degrés
    avancer de 50 pas
    tourner de 90 degrés
  
```

