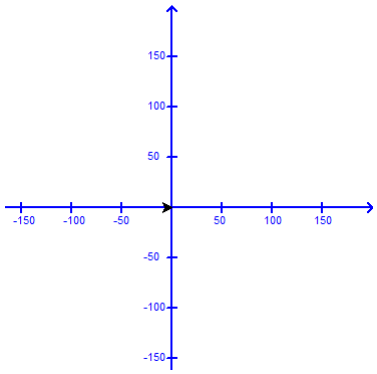
Principales méthodes pour `turtle`

Les instructions du tracé sont affectés au crayon défini par défaut.

Le crayon utilisé pour les tracés est un **objet** informatique.
On peut donc aussi définir (*nommer*) un crayon grâce à l'instruction d'affectation.



Par exemple, l'instruction :

```
c = Pen()
```

affecte un crayon (une tortue) à la variable nommée `c`.

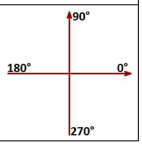
Pour manipuler cet **objet**, il faut ensuite lui appliquer une **méthode** selon la syntaxe :

`nom_crayon.nom_methode(parametres eventuels)`

I – Comportement du crayon

Méthode(paramètre)	Description
`shape(forme)`	`forme` possible : `'classic'` ou `'turtle'`
`speed(n)`	Vitesse du tracé : de `n = 1` (*lent*) à `n = 10` (*rapide*).
`width(n)`	Épaisseur du trait : de `n = 1` (*fin*) à `n = 10` (*épais*).
`up()`	Relève le crayon (*pour le déplacer sans dessiner*).
`down()`	Abaisse le crayon (*pour recommencer à dessiner*).
`home()`	Ramène le crayon dans sa position initiale.
`hideturtle()`	Masque le crayon.

II – Déplacements usuels du crayon

Méthode(paramètre)	Description
`forward(distance)`	Avance d'une `distance` (*en pixels*) donnée.
`backward(distance)`	Recule d'une `distance` (*en pixels*) donnée.
`goto(x,y)`	Déplace le crayon jusqu'aux coordonnées `(x;y)`.
`left(angle)`	Tourne à gauche d'un `angle` donné (*en degrés*).
`right(angle)`	Tourne à droite d'un `angle` donné (*en degrés*).
`setheading(angle)`	Pointe le crayon vers la direction (*absolue*) indiquée par l'`angle`. Cet `angle` est donné en degrés 

III – Les couleurs

Méthode(paramètre)	Description
`colormode(n)`	Initialise le format `RGB`, `n` est l'intensité maximale.
`pencolor(couleur)`	`couleur` du tracé, de type `string` ou `(r, g, b)`
`fillcolor(couleur)`	Permet de remplir un contour fermé avec la `couleur`. `begin_fill()` doit être indiqué **avant le tracé** du contour fermé, `end_fill()` **après**.
`color(couleur1, [couleur2])`	Définit la `couleur1` du tracé puis la `couleur2` du remplissage. Si `couleur2` n'est pas spécifié, `couleur1` s'applique aux deux.

IV – Tracés spécifiques

Méthode(paramètre)	Description
`write(texte)`	Le **`texte`**, de type **`string`**, est écrit à la position actuelle du crayon (*avec la couleur courante*).
`circle(x,[y])`	Trace un **cercle** de rayon **`x`**, dans la continuité du tracé précédent. Possibilité d'**arc de cercle** avec une valeur d'angle **`y`** (*sans crochets pour l'utiliser*).
`dot(x,[couleur])`	**Disque** de diamètre **`x`**, centré à l'endroit où se trouve le crayon (***`couleur`*** *intérieure en option*).

V – Récupérer des informations sur le crayon

Méthode(paramètre)	Description
`position()`	Renvoie les coordonnées actuelles du crayon (*tuple*).
`heading()`	Renvoie l'orientation (*en degrés*) actuelle du crayon.

VI – Agir sur la fenêtre d'affichage

Méthode(paramètre)	Description
`clear()`	Efface le dessin, le crayon reste à sa place.
`reset()`	Ré-initialise la page (*dessin effacé, crayon à l'origine*).
`setworldcoordinates(xbg, ybg, xhd, yhd)`	Redéfinition du système de coordonnées dans une nouvelle fenêtre définie par les points en bas à gauche **`(xbg, ybg)`** et en haut à droite **`(xhd, yhd)`**.
`exitonclick()`	Permet de sortir du script en cliquant sur la fenêtre. **A écrire en fin de script**.

VII – Interaction avec l'utilisateur

Les instructions ci-dessous affichent une fenêtre « popup » dont le nom est **`Nom_Fenetre`** avec un **`Message`** écrit à destination de l'utilisateur. L'utilisateur saisi une valeur dans le champs de texte, valeur qui est stockée dans la variable **`a`**.

Instruction	Description
`a = textinput('Nom_fenetre', 'Message')`	**`a`** est de type **`string`**.
`a = numinput('Nom_fenetre', 'Message')`	**`a`** est de type **`float`**.

VIII – Remarque

Avec certains IDE, il est parfois nécessaire d'insérer l'instruction **`TurtleScreen._RUNNING = True`** dans le programme principal pour éviter des « plantages ».

De plus, il est conseillé de terminer le programme par l'instruction **`mainloop()`**, issue du module **`tkinter`**, et qui permet d'initialiser « l'écouteur » d'événements.