

## A. Des instructions disponibles directement

|   | Syntaxe                    | Exemple   | Syntaxe                     | Exemple  |
|---|----------------------------|---|-----------------------------|--|
| addition  | +                          | <code>&gt;&gt;&gt; 2 + 5<br/>7</code>   | division                    | <code>/</code><br><code>&gt;&gt;&gt; 14 / 3<br/>4.6666666666666667</code>  |
| soustraction                                      | -                          | <code>&gt;&gt;&gt; 2 - 5<br/>-3</code>  | arrondir x à $10^{-n}$ près | <code>round(x,n)</code><br><code>&gt;&gt;&gt; round(2.56,1)<br/>2.6</code> |
| multiplication                                    | *                          | <code>&gt;&gt;&gt; 2 * 5<br/>10</code>  | maximum                     | <code>max(a,b,...)</code><br><code>&gt;&gt;&gt; max(2, 5, 3)<br/>5</code>  |
| puissance   | **                         | <code>&gt;&gt;&gt; 2 ** 5<br/>32</code>   | minimum                     | <code>min(a,b,...)</code><br><code>&gt;&gt;&gt; min(2, 5, 3)<br/>2</code>  |
| quotient<br>et reste dans la division euclidienne | // (quotient)<br>% (reste) | <code>&gt;&gt;&gt; 14 // 3<br/>4</code><br><code>&gt;&gt;&gt; 14 % 3<br/>2</code> | = et ≠<br>== et !=          | <code>&gt;&gt;&gt; 21 / 2 == 10.5<br/>True</code>                          |
|   |                            |   | < , ><br>≤ , ≥              | <code>&gt;&gt;&gt; 22 / 7 &lt; 3<br/>False</code>                          |

## B. Avec les bibliothèques math ou random

|  |  |
|--|--|
|  | <code>&gt;&gt;&gt; from math import*</code>                        |
| Nombre $\pi$ .   | <code>&gt;&gt;&gt; pi<br/>3.141592653589793</code>                 |
| Racine carrée d'un nombre.   | <code>&gt;&gt;&gt; sqrt(2)<br/>1.4142135623730951</code>           |
| Partie entière d'un nombre.  | <code>&gt;&gt;&gt; floor(5.72)<br/>5</code>                        |
| Cos, sin, tan d'un angle mesuré en degrés.   | <code>&gt;&gt;&gt; cos(radians(30))<br/>0.8660254037844387</code>  |
| Angle en degrés connaissant son cosinus, sinus ou sa tangente (analogues de $\cos^{-1}$ , $\sin^{-1}$ , $\tan^{-1}$ ). | <code>&gt;&gt;&gt; degrees(acos(0.8))<br/>36.86989764584401</code> |

|  |  |
|--|--|
|  | <code>&gt;&gt;&gt; import random</code>  |
| Nombre entier aléatoire de l'intervalle $[a,b]$ (de $[1 ; 6]$ dans l'exemple). Analogue de ALEA.ENTRE.BORNES() sur le tableur. | <code>&gt;&gt;&gt; random.randint(1, 6)<br/>4</code>   |
| Nombre réel au hasard dans l'intervalle $[0 ; 1[$ . Analogue de ALEA() sur le tableur.   | <code>&gt;&gt;&gt; random.random()<br/>0.25236135296415485</code>                            |
| Un élément tiré au hasard dans une liste L.  | <code>&gt;&gt;&gt; L=[0, 5, 10, 15, 20]<br/><br/>&gt;&gt;&gt; random.choice(L)<br/>10</code> |

## C. Avec la bibliothèque turtle

On importe la bibliothèque par `import turtle` pour avoir accès aux commandes ci-après (et à beaucoup d'autres).

|  | <b>Scratch</b>   | <b>Actions</b>   | <b>Python</b>   |
|--|--|--|---|
| <b>Gestion du crayon et de la tortue</b> | <b>cacher</b>  | Cacher la tortue.  | <code>turtle.hideturtle()</code>                                    |
|  | <b>montrer</b>   | Montrer la tortue.   | <code>turtle.showturtle()</code>                                    |
|  | <b>effacer tout</b>  | Effacer tout.  | <code>turtle.clear()</code>   |
|  | <b>effacer tout</b><br><b>aller à x: 0 y: 0</b><br><b>s'orienter à 90°</b> | Effacer tout, recentrer la tortue, l'orienter vers la droite Les variables sont aussi remises à 0.                         | <code>turtle.reset()</code>   |
|  | <b>stylo en position d'écriture</b>  | Abaïsser le crayon pour pouvoir dessiner.  | <code>turtle.down()</code>  |
|  | <b>relever le stylo</b>  | Relever le crayon.   | <code>turtle.up()</code>  |
|  | <b>mettre la couleur du stylo à</b> [color]                                | Choisir la couleur du trait.*  | <code>turtle.pencolor(couleur)</code>                               |
|  | <b>mettre la taille du stylo à</b> [width]                                 | Choisir l'épaisseur du trait.  | <code>turtle.width(epaisseur)</code>                                |
|  |  | Remplir la forme qui va être tracée/<br>Arrêter le remplissage.  | <code>turtle.begin_fill(couleur choisie) / turtle.end_fill()</code> |
| <b>Mouvement</b>                         |  | Écrire un texte à la position de la tortue.  | <code>turtle.write(texte)</code>                                    |
|  | <b>s'orienter à 90°</b>  | Orienter la tortue.  | <code>turtle.setheading(0)<br/>(0 ou 90 ou 180 ou 270)</code>       |
|  | <b>aller à x: -100 y: -100</b>   | Aller à la position de coordonnées (x,y).  | <code>turtle.goto(x,y)</code>                                       |
|  |  | Obtenir la position de la tortue.  | <code>turtle.position()</code>                                      |
|  | <b>avancer de 10</b>   | Avancer d'un certain nombre de pas.  | <code>turtle.forward(nombre de pas)</code>                          |
|  | <b>avancer de -10</b>  | Reculer d'un certain nombre de pas.  | <code>turtle.back(nombre de pas)</code>                             |
|  | <b>tourner ⚡ de 15 degrés</b>  | Tourner à droite d'un angle donné.   | <code>turtle.right(mesure de l'angle)</code>                        |
|  | <b>tourner ⚡ de 15 degrés</b>  | Tourner à droite d'un angle donné.   | <code>turtle.left(mesure de l'angle)</code>                         |
|  |  | Suivre un arc de cercle à partir de sa position de rayon r, d'un angle a (positif ou négatif selon la position du centre). | <code>turtle.circle(r,a)</code>                                     |
|  |  | Choisir la vitesse de la tortue : speed(1) lent, speed(10) rapide.   | <code>turtle.speed(nombre entre 1 et 10)</code>                     |

\* Couleurs prédéfinies : 'blue', 'red', 'green', 'yellow', 'orange', 'purple', 'brown', 'black', 'white', 'pink', 'grey'

À l'exécution, une fenêtre graphique s'ouvre (elle peut être réduite dans la barre des tâches). On finira le programme par l'instruction `turtle.exitonclick()` et il suffira de cliquer sur la fenêtre graphique pour la fermer.

### Remarques

- un bug subsiste dans certaines versions de Python quand on relance le programme... il suffit de le relancer une fois de plus.
- on peut travailler avec plusieurs tortues auxquelles on peut donner des noms pour les identifier.

## D. Avec matplotlib.pyplot

On choisit de créer un alias (plt) pour raccourcir l'écriture et faciliter la lisibilité des instructions.

|               |  | >>> <b>import matplotlib.pyplot as plt</b>  |
|---------------|--|---|
| <b>Repère</b> | Choisir un repère orthonormé.  | plt.axis('equal')   |
|               | Choisir [xmin, xmax, ymin, ymax].  | plt.axis([0, 6, 0, 20])   |
|               | Afficher la grille.  | plt.grid()  |
|               | Indiquer une légende sur l'axe des abscisses ou des ordonnées.                                       | plt.xlabel('temps en h')<br>plt.ylabel('distance en km')                          |
| <b>Points</b> | Placer le point A (xA ; yA).   | plt.plot([xA], [yA])  |
|               | Marquer A(xA ; yA) par un petit point rouge.   | plt.plot([xA ; yA], 'r.')   |
|               | Marquer A(xA ; yA) par un gros point bleu.   | plt.plot([xA], [yA], 'bo')  |
|               | Marquer A(xA ; yA) par un gros carré bleu (bs = blue square).  | plt.plot([xA], [yA], 'bs')  |
|               | Marquer B(xB ; yB) par un triangle vert (g^ = green triangle).                                       | plt.plot([xB], [yB], 'g^')  |
|               | Marquer C(xC ; yC) par un trait en pointillés rouge.   | plt.plot([xC], [yC], 'r--')   |
| <b>Ligne</b>  | Placer les points A(xA ; yA), B(xB ; yB), C(xC ; yC) et D(xD, yD).                                   | plt.plot([xA, xB, xC, xD], [yA, yB, yC, yD])                                      |
|               | Tracer le segment [AB].  | plt.plot([xA, xB], [yA, yB], '-')   |
|               | Tracer le segment [AB] en rouge.   | plt.plot([xA, xB], [yA, yB], 'r-')  |
|               | Créer une liste vx de valeurs de t.<br>Placer les points (t, t <sup>2</sup> ) pour créer une courbe. | vx = [t/10 for t in range(-5, 55)]<br>vy = [t**2 for t in vx]<br>plt.plot(vx, vy) |
| <b>Figure</b> | Afficher la figure.  | plt.show()  |
|               | Créer différentes figures.   | plt.figure(1)<br><i>instructions</i><br>plt.figure(2)<br><i>instructions</i>      |

### Exemple

```

1 import matplotlib.pyplot as plt
2
3 plt.grid()
4 # liste des nombres de 0 à 5 avec un pas de 0,1
5 vx = [t/10 for t in range(0, 51)]
6 # liste des carrés des nombres de la liste vx
7 vy = [t**2 for t in vx]
8 plt.plot(vx, vy, 'r-')
9 plt.plot([0, 5], [0, 25], 'b--')
10 plt.show()

```

