

## Aide-mémoire Python

| <b>Variables</b>   |   |
|--|---|
| <pre>x = 1 + 2 x = x + 1 x += 1</pre>  | déclare une variable <code>x</code> , initialisée avec <code>1+2</code><br>incrémente la variable <code>x</code><br>une autre façon de l'écrire   |
| <b>Entrées-sorties</b>   |   |
| <pre>print(2 * x) print('texte') s = input('votre nom :') n = int(input('entrez n :'))</pre> | affiche la valeur de <code>2 * x</code><br>affiche la chaîne <code>texte</code><br>affiche <code>votre nom :</code> et attend une saisie au clavier, stockée dans la variable <code>s</code><br>attend une saisie au clavier et l'interprète comme un entier, stocké dans la variable <code>n</code>  |
| <b>Entiers</b>   |   |
| <pre>n // 3 n % 3 n == 17 n != 17</pre>  | le quotient de la division euclidienne de <code>n</code> par <code>3</code><br>le reste de la division euclidienne de <code>n</code> par <code>3</code><br>est-ce que <code>n</code> est égal à <code>17</code> ? (booléen)<br>est-ce que <code>n</code> est différent de <code>17</code> ? (booléen) |
| <b>Conditionnelle</b>  |   |
| <pre>if n &gt; 100:     n = n - 10 else:     n = n + 11</pre>                                | si <code>n</code> est plus grand que <code>100</code><br>alors lui retrancher <code>10</code><br>sinon<br>lui ajouter <code>11</code>   |
| <b>Boucle sur les entiers</b>  |   |
| <pre>for n in range(0, 100):     print(n)</pre>  | parcourt tous les entiers de <code>0</code> inclus à <code>100</code> exclus<br>la variable <code>n</code> prend successivement chaque valeur   |
| <b>Boucle tant que</b>   |   |
| <pre>while n &lt; 1000:     n = 2 * n</pre>  | tant que <code>n</code> est plus petit que <code>1000</code><br>multiplier <code>n</code> par <code>2</code>  |
| <b>Hasard</b>  |   |
| <pre>from random import randint randint(0, 10)</pre>   | on importe la fonction <code>randint</code> de la bibliothèque <code>random</code><br>un entier tiré au hasard entre <code>0</code> et <code>10</code> inclus   |

| <b>Tableaux</b>   |  |
|---|--|
| <pre>t = [0, 1, 1, 2, 3, 5] t = [0] * 100  t[3]  t[2] = 42 len(t)  for x in t:     print(x)</pre>             | <p>déclare un tableau <b>t</b> contenant 6 entiers<br/>déclare un tableau <b>t</b> de taille 100, toutes les cases contenant 0</p> <p>la valeur contenue dans la quatrième case du tableau <b>t</b> (les cases sont numérotées à partir de 0)</p> <p>modifie la troisième case du tableau</p> <p>le nombre d'éléments du tableau <b>t</b> (de l'anglais <i>length</i> pour longueur)</p> <p>parcourt le tableau <b>t</b><br/>on fait quelque chose avec <b>x</b>, ici l'afficher</p> |
| <b>Ensembles</b>  |  |
| <pre>s = set() s.add(42) 89 in s  len(s) for x in s:     print(x)</pre>                                       | <p>déclare un ensemble <b>s</b>, vide<br/>ajoute l'entier 42 dans l'ensemble <b>s</b><br/>est-ce que 89 est dans l'ensemble <b>s</b>? (booléen)</p> <p>le nombre d'éléments de l'ensemble <b>s</b><br/>parcourt l'ensemble <b>s</b><br/>on fait quelque chose avec <b>x</b>, ici l'afficher</p>  |
| <b>Dictionnaires</b>  |  |
| <pre>d = {} d['toto'] = 42 'toto' in d d['titi'] len(d) for x, v in d.items():     print(x, '-&gt;', v)</pre> | <p>déclare un dictionnaire <b>d</b>, vide<br/>associe l'entier 42 à la chaîne <b>'toto'</b><br/>est-ce que <b>'toto'</b> est une clé? (booléen)</p> <p>la valeur associée à la chaîne <b>'titi'</b><br/>le nombre d'entrées dans le dictionnaire <b>d</b><br/>parcourt les clés/valeurs de <b>d</b><br/>affiche ici toutes les entrées</p>   |

### Turtle

```
from turtle import *
```

```
goto(100,50)
```

```
forward(60)
```

```
backward(40)
```

```
left(45)
```

```
right(120)
```

```
circle(10, 30)
```

```
up()
```

```
down()
```

```
color(0.2, 0.2, 0.2)
```

```
begin_fill()
```

```
end_fill()
```

on importe toutes les fonctions de la bibliothèque `turtle`

`goto(100,50)` va au point de coordonnées (100, 50)

`forward(60)` avance de la distance 60

`backward(40)` recule de la distance 40

`left(45)` pivote à gauche de 45 degrés

`right(120)` pivote à droite de 120 degrés

`circle(10, 30)` trace un arc de cercle de 30 degrés et de rayon 10

`up()` lève le crayon (ne dessine plus)

`down()` pose le crayon (dessine de nouveau)

`color(0.2, 0.2, 0.2)` fixe la couleur, avec des composantes rouge/vert/bleu entre 0 et 1

`begin_fill()` commence un tracé qui sera rempli

`end_fill()` remplit ce qui a été tracé depuis le dernier appel à `begin_fill()`