

Fiche de révision – Algorithmique et Python

À relire avant le contrôle. Coche ce que tu sais faire, retravaille le reste avec les exercices et le quiz.

L'essentiel

- Types : `int` (5), `float` (2.5), `str` ("abc"), `bool` (True).
- Affectation `x = 3`; test d'égalité `x == 3`.
- `//` quotient entier; `%` reste; `**` puissance.
- `for i in range(n)` : n tours, i de 0 à $n-1$.
- `while condition`: répète **tant que** la condition est vraie.
- `return` renvoie une valeur; `print` l'affiche.

Méthodes et astuces

- Nombre de tours **connu** : `for`; **inconnu** : `while`.
- Faire un **tableau d'état** des variables pour suivre un programme.
- Un compteur : `c = 0` avant la boucle, `c = c + 1` dedans.
- Une somme : `s = 0` avant, `s = s + ...` dedans.
- Ne pas oublier les `:` et l'indentation.

Rédaction modèle

Plus petit n tel que $2^n > 1000$.

```
n = 0
while 2**n <= 1000:
    n = n + 1
print(n)    # 10
```

Pièges classiques

- ✗ `range(1, 10)` s'arrête à 9.
- ✗ `"5" + "3"` donne "53", pas 8.
- ✗ `if x = 3:` : il faut `==`.
- ✗ Boucle `while` infinie si la condition ne devient jamais fausse.
- ✗ `7/2` donne 3.5 mais `7//2` donne 3.

Modèles à connaître

```
def f(x):
    return x**2 - 3*x + 1

from random import randint
de = randint(1, 6)

s = 0
for k in range(1, 11):
    s = s + k
```

Je sais faire

- ☐ Donner le type d'une variable.
- ☐ Suivre un programme avec un tableau d'état.
- ☐ Écrire une condition `if/elif/else`.
- ☐ Écrire une boucle `for` et une boucle `while`.
- ☐ Écrire et appeler une fonction.
- ☐ Simuler une expérience aléatoire.

Mes notes : ce que je dois retravailler, mes questions pour le professeur