

AFFECTATION

1 Algorithme, variable

▣ Définition 1

- Un **algorithme** est une suite d'instructions permettant de résoudre un problème donné en un nombre fini d'opérations. Il peut être implémenté en un programme après traduction dans un langage de programmation.
- Une **variable** est un symbole reliant un nom à une valeur. On dit qu'une variable **stocke** une valeur ; l'attribution d'une valeur à une variable est désigné par « **affectation** ».
- On appelle **entrée** une variable dont l'utilisateur peut choisir la valeur lors de l'exécution de l'algorithme.
- On appelle **sortie** une variable qui sera récupérée après exécution de l'algorithme.

↪ Exemple 1

↻ Cet algorithme convertit une température c donnée en degré CELSIUS choisie par l'utilisateur en sa valeur f en degré FAHRENHEIT :

- enlever 32 à c ;
- puis diviser la valeur obtenue par 1,8.

▣ Définition 2

Choix a été fait de présenter les algorithmes sous forme d'un « **langage naturel** », permettant d'harmoniser les présentations grâce à certaines conventions

- on présente en préambule la ou les entrée(s) et la ou les sortie(s) à l'aide de phrases ;
- l'affectation de la valeur x à la variable v est symbolisée par « $v \leftarrow x$ » ;
- les opérations usuelles (addition, soustraction, multiplication, division/fraction, exposants, racine carrée, ...) sont présentées usuellement.

↪ Exemple 2

↻ Cet algorithme a pour entrée une valeur c , température donnée en degré CELSIUS et renvoie f , sa valeur en degré FAHRENHEIT.

$$f \leftarrow \frac{c - 32}{1,8}$$

2 Python

Python est un langage de programmation inscrit au programme de la seconde.

2.1 Opérations

□ Définition 3

Pour deux nombres a et b

Somme de a et b	$a+b$	« plus »
Différence de a avec b	$a-b$	« tiret court »
Produit de a et b	$a*b$	« astérisque »
a exposant b	$a**b$	« double astérisque »
Quotient (décimal) de a par b	a/b	« barre oblique » (slash)
Quotient par division euclidienne de a par b	$a//b$	« double barre oblique » (double slash)
Reste par division euclidienne de a par b	$a\%b$	« pourcentage »

N.B. : on peut accéder à la dernière valeur évaluée *via* le tiret bas (« barre du 8, underscore ») : `_`.

□ Définition 4

On affecte à une variable une valeur (numérique ou non) grâce à `=`.

N.B. : on peut donner des noms explicites aux variables - non limité à une lettre - et c'est même conseillé par soucis de lisibilité.

2.2 Typage

2.2.1 Ensembles de nombres

□ Définition 5

- Un **entier relatif** est un entier positif ou négatif. L'ensemble de ces nombres est noté \mathbb{Z} .
- Un **nombre décimal** est un nombre pouvant s'écrire sous la forme d'une fraction de deux entiers, avec un dénominateur sous la forme d'une puissance de dix (10^a). Un tel nombre a sa partie décimale finie (au-delà d'un certain rang, toutes les décimales sont 0). L'ensemble de ces nombres est noté \mathbb{D} .
- Un **nombre rationnel** est un nombre pouvant s'écrire sous la forme d'une fraction de deux entiers. L'ensemble de ces nombres est noté \mathbb{Q} .
- Un **nombre irrationnel** est un nombre qui ne peut pas s'écrire sous la forme d'une fraction de deux entiers.

↪ Exemple 3

- -2 ; 0 ; $2\ 017$ sont des entiers relatifs, mais aussi des nombres décimaux, rationnels, réels. On peut aussi noter : $-2 \in \mathbb{Z}$; $2\ 017 \in \mathbb{R}$.
- $\frac{46}{20} = \frac{23}{10} = 2,3 \in \mathbb{D}$ et $2,3 \notin \mathbb{Z}$.
- $\frac{1}{3} \in \mathbb{Q}$ et $\frac{2}{7} \notin \mathbb{D}$.
- π et $\sqrt{2}$ sont des nombres irrationnels.

2.2.2 Les types à connaître

▣ Définition 6

Le programme Python attribue un type aux variables et objets ; voici quelques types :

- le type **entier** (integer) qui correspond aux nombres entiers relatifs ;
- le type **flottant** (float), souvent représenté en notation scientifique : $M \times 10^e$ (une mantisse M , nombre décimal compris entre -10 et 10 , et un exposant entier e) qui correspond aux nombres décimaux ;
- le type **chaîne de caractère** (string), qui permet d'implémenter des messages textuels ;
- le type **list** (liste) utile en statistique et pour réaliser des graphiques.

↪ Exemple 4

- `0`, `-71`, `1024` sont des nombres considérés *a priori* comme entiers (l'espace séparateur des milliers n'est pas réalisé).
- `6.9`, `2.0`, `6.022E23` sont des nombres considérés comme flottants (le séparateur décimal est le point et `6.022E23` ↔ $6,022 \times 10^{23}$).
- `'doing maths is fun'`, `'cats are cool'`, `'#420 is bad'` sont des chaînes de caractères.

N.B. : on peut connaître le type d'un objet en faisant `type(objet)`.

2.2.3 Forcer un typage

▣ Définition 7

Des variables peuvent être changées de type avec les commandes

- `int()` pour convertir en entier ;
- `float()` pour convertir en flottant ;
- `str()` pour convertir en chaîne de caractère.

2.3 Objet intégré input

▣ Définition 8

L'utilisateur peut choisir la valeur d'une variable grâce à l'objet intégré `input()` : elle jouera le rôle d'entrée dans un module Python.

Par défaut, cette variable est traitée en chaîne de caractère, mais on peut forcer son typage au besoin.

↪ Exemple 5

Ce module Python affiche le mot choisi répété (« concaténé ») par la valeur choisie.

```
valeur=int(input('Quelle nombre ?'))
mot=str(input('Quel mot ?'))
print(valeur*mot)
```

3 Fonction informatique

▣ Définition 9

Une **fonction informatique** est une suite d'instructions aboutissant au calcul d'une variable de sortie. Une telle fonction peut admettre des variables d'entrée, appelées **arguments**.

Une fonction est définie ainsi sur Python

```
def nomfonction(entrée) :
    effet
    return sortie
```

- le mot clef **def** annonce la définition de la fonction.
- Le bloc définissant la fonction est en retrait du reste du code : on appelle ce retrait **l'indentation**. Il sert à mettre en évidence certaines parties du code.
- La variable sortie est annoncée avec le mot clef **return**.
- **nomfonction(entrée)** renvoie la variable de sortie de la fonction **nomfonction** pour l'argument **entrée**.

↪ Exemple 6



On définit ainsi la fonction f d'expression $f(x) = x^2 - x - 1$, et on calcule ensuite $f(2)$.

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

f(2)
```

4 Évolution

▣ Définition 10

Pour une grandeur évoluant d'une valeur de départ, notée V_D , vers une valeur d'arrivée, notée V_A , le **taux d'évolution** de V_D vers V_A est

$$\tau = \frac{V_A - V_D}{V_D}.$$

≠ Propriété 1

Pour une grandeur évoluant d'une valeur de départ, notée V_D , vers une valeur d'arrivée, notée V_A , au taux d'évolution τ , on a la relation

$$V_A = (1 + \tau)V_D.$$

N.B. : la valeur $1 + \tau$ est appelé **coefficient multiplicateur** associé au taux τ .

↪ Exemple 7



Retrouver la valeur manquante (V_D ; τ ou V_A) dans chaque situation.

1. Une population d'éléphants de mer passe de 150 à 175 individus.
2. Un pantalon possède deux étiquettes de prix : 70 € et -30 %.
3. Après une diminution du capital de 60 %, le capital ne s'élève plus qu'à 7 000 €.
4. Un particulier change son ordinateur : son processeur passe d'une cadence de 1,54 GHz à 4,86 GHz.
5. Une population d'ocelots diminue de 89 % : il n'en reste plus que 44.
6. La part d'huile de palme dans la consommation mondiale était de 11 % en 1980 : elle a bondi de 209 %.