

Fiche récapitulative Algo & Base Python

Algorithmme, Variables et affectations

Variables

Def : “Une variable est un lieu de stockage portant un nom”

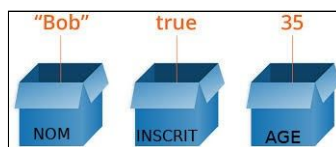
Note : Le nom d’une variable peut être composé de :

- lettres (majuscules ou minuscules) sans accents
- chiffres
- le caractère ‘_’ (qui remplace les espaces)

Opération & Affectations

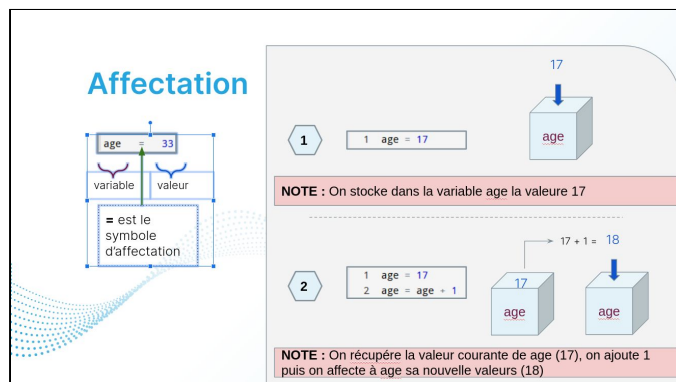
Def : “Une affectation est la transmission d’une donnée (valeurs) à une variable.”

```
3 nom = "bob"
4 inscrit = True
5 age = 35
```



4 types de variables :

- **Entier (int)** : Nombre positif ou négatif (sans virgule). Exemple : -8974 ; -3 ; 0 ; 42; 3025
- **Flottant (float)** : Nombre à virgule. Exemple : -0.3 ; -15.556 ; 1.72 ; 3.14
- **Chaîne de caractères (string)** : Texte délimité par des guillemets. Exemple : “toto”, “azerty123”, “John”
- **Booléen (boolean)** : Binaire Vrai ou Faux. Exemple True/False



Note : L’addition (+) concatène (colle) les String (exemple “Hello” + “ “ + “World” = “Hello World”)

Algorithmme

Def1 : “Ensemble de règles opératoires dont l’application permet de résoudre un problème énoncé au moyen d’un nombre fini d’opérations.” *Larousse 2019*

Def2 : “Ecrire un algorithme, c’est dessiner un chemin de résolution pour un problème donné” *Aurélie Jean*

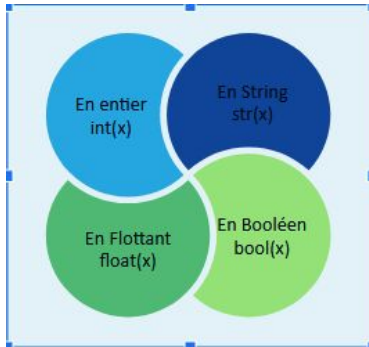
Exemple d’algorithmes : Reconnaissance faciales, résolution de labyrinthe, compression d’image...

Python Bases

```
1 a = 1
2 # Le commentaires sont ignorés à l'exécution
3 # a = 52
4 print(a) # Affiche 1
```

Note : Le symbole ‘#’ permet de commenter une ligne

Conversions



```
1 a = 3
2 b = "5"
3
4 # -> Erreur lors de l'exécution
5 c = a + b
6
7 # Ok car du même type (entier)
8 c = a + int(b)
```

NOTE : Les opérations (+,-,*,/...) sur des types différents sont à éviter en Python !

Comparateurs

	Signification	Exemples
<code>==</code>	Egalité	<code>x == 0</code> , <code>date == 2021</code> , <code>user == "toto"</code>
<code>!=</code>	Différent	<code>diviseur != 0</code> ;
<code>></code> , <code><</code>	Strictement Supérieur, Inférieur	<code>age > 18</code> ; <code>temperature < 0</code>
<code>>=</code> , <code><=</code>	Égale ou Supérieur, Inférieur	<code>age >= 18</code> ; <code>date >= 1963</code>

```
1 age = 17
2 nom = "Alan"
3
4 # Exemple d'égalités/différences
5 print(age == 17)      # Affiche True
6 print(age == 18)      # Affiche False
7
8 print(nom == "Alan")  # Affiche True
9 print(nom == "Steve") # Affiche False
10 print(nom == "ALAN") # Affiche ???
```

Note : Les comparateurs renvoient un Booléen ! (Vrai ou Faux)

```
6 age = 18 # la variable age vaut maintenant 18
7 age == 18 # test si age est égale à 18
```

Note : Le comparateur d'égalité (`==`) ne doit pas être confondu avec l'affectation (`=`).

Entrée/Sortie

Sortie

L'instruction `print` permet d'afficher du texte ou le contenu d'une variable à l'écran.

```
1 print("Hello world")
2 age_marc = 17
3 print(age_marc)
4 age_marc = age_marc + 1
5 print("Marc est maintenant majeur, il a : " + str(age_marc))
```

Entrée

L'instruction `input` attend que l'utilisateur tape un texte. Il faut affecter le contenu du `input` à une variable ! Par défaut un `input` considère l'entrée comme une chaîne de caractère.

```
1 prenom = input("Comment vous appelez-vous ?")
2 print("Bonjour" + prenom)
3 age = int( input("Quel age avez-vous ?") )
4 print("Vous avez : " + str(age))
```

Note : Penser à convertir l'entrée si vous souhaitez autre chose qu'un String !