



BAC Pro CIEL	<h2>Python – Module 1</h2> <h3>Bases de la programmation</h3>	 Année 2025/2026
		

A. Les variables

Une variable est un élément commun et indispensable à tous les langages de programmation permettant de stocker des informations dans la mémoire vive d'un ordinateur, d'un microcontrôleur ou une puce électronique mémoire.

Ces informations stockées peuvent être des nombres, du texte, des dates, etc...

En Python, il n'est pas nécessaire de déclarer le type de variable comme dans d'autres langage de programmation (C, C++), Python interprète et devine automatiquement le type de variable en fonction de ce que tu mets dedans : le typage dynamique.

Tout d'abord, il est indispensable de donner un nom à la variable, ce nom lui servira d'identifiant. Par exemple on peut avoir des noms comme age, nom_utilisateur, temperature, position, ...

Quelques règles importantes pour les noms de variables :

- ⇒ Elles doivent commencer par une lettre ou un underscore (_)
- ⇒ Elles ne peuvent contenir que des lettres, chiffres et underscores
- ⇒ Elles sont sensibles à la casse (Age est différent de age)
- ⇒ Il faut éviter d'utiliser les mots-clés réservés de Python (if, for, print, while, ...)

Ensuite pour affecter une donnée (valeur) à une variable on utilise le signe égal (=).

Exemple :

```
age = 18           # La variable age contient le nombre 18
nom = "Alice"     # La variable nom contient le texte « Alice »
prix_ht = 15.50   # La variable prix_ht contient le nombre 15.50
est_la = True     # La variable est_la contient le booléen True
```

Les variables permettent donc de stocker des données et permettent de pouvoir les réutiliser dans la manipulation de données. Par exemple :

```
cpt = 0           # La variable cpt contient le nombre 0
cpt = cpt + 1     # La variable cpt contient maintenant le nombre 1
```

B. Afficher

La fonction `print()` en Python est une fonction intégrée essentielle utilisée pour afficher des informations sur la console. C'est l'un des moyens les plus courants de déboguer du code et de fournir une sortie à l'utilisateur.

La syntaxe la plus simple est :

```
print(objet_a_afficher)
```

avec `objet_a_afficher` peut être n'importe quel type de donnée (chaîne de caractères, nombre, liste, etc...).

Code Python	Résultat
<code>print("Bonjour")</code>	Bonjour
<code>print("salut")</code> <code>print("Marc")</code>	salut Marc
<code>print("Bonjour tout le monde")</code>	Bonjour tout le monde
<code>print("Salut","toi")</code>	Salut toi
<code>print("Salut"+"toi")</code>	Saluttoi
<code>print(192)</code>	192
<code>print("192")</code>	192
<code>print(3.14)</code>	3.14
<code>print("3.14")</code>	3.14
<code>nourriture = "pomme"</code> <code>print("La",nourriture)</code>	La pomme
<code>fruit = "banane"</code> <code>print("La",fruit,"est bonne")</code>	La banane est bonne
<code>prix = 2.26</code> <code>print("Le tarif est",prix)</code>	Le tarif est 2.26
<code>prix = 1.31</code> <code>legume = "carotte"</code> <code>print("Le prix de",legume,"est de",prix)</code>	Le prix de carotte est de 1.31
<code>print("pomme","banane","orange",sep="-")</code>	pomme-banane-orange
<code>print("jean","lafitte",sep="@")</code>	jean@lafitte
<code>print("Un",end="...")</code> <code>print("Deux")</code>	Un...Deux
<code>genre = "chat"</code> <code>poids = 6.25</code> <code>print(f"Le {genre} pèse {poids} kg.")</code>	Le chat pèse 6.25 kg.
<code>objet = "PC"</code> <code>ville = "Loches"</code> <code>print("Ce {} vient de {}".format(objet,ville))</code>	Ce PC vient de Loches.
<code>print("1ere Ligne \n2eme Ligne\n 3eme Ligne")</code>	1ere Ligne 2eme Ligne 3eme Ligne
<code>print("Ecrire un Guillemet \" facile !")</code>	Ecrire un Guillemet " facile !

C. Saisie utilisateur

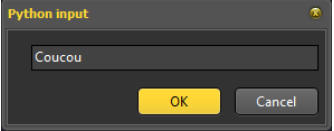
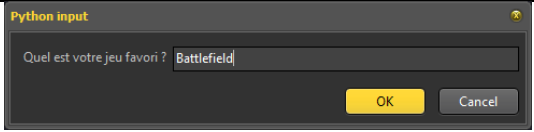
La fonction `input()` est une fonction intégrée qui permet à votre programme de demander une saisie à l'utilisateur via la console. Le programme s'arrête de s'exécuter jusqu'à ce que l'utilisateur tape quelque chose et appuie sur la touche Entrée.

La syntaxe de base est :

```
variable = input(message_invite)
```

avec `message_invite`, la chaîne de caractère qui sera affichée à l'utilisateur avant qu'il ne saisisse sa réponse. Cette option est facultative, si elle n'est pas utilisée, laisser le champ vide.

avec `variable`, la valeur saisie par l'utilisateur après avoir appuyé sur Entrée. Elle est renvoyée sous forme de chaîne de caractères et stockée dans cette variable.

Code Python	Résultat
<pre>msg_utilisateur = input() print(f"Vous avez entré : {msg_utilisateur}")</pre>	 Vous avez entré : Coucou
<pre>msg = input("Quel est votre jeu favori ?") print(f"Votre jeu préféré est {msg}")</pre>	 Votre jeu préféré est Battlefield

D. Convertir un texte en nombre

Soit la variable « mon_texte » suivante :

```
mon_texte = "24"
```

Dans ce cas, cette variable possède en mémoire une chaîne de caractère et non un nombre. Il est donc impossible d'y effectuer des calculs numériques dessus. Il est donc nécessaire d'effectuer une conversion :

```
mon_nombre = int(mon_texte)
```

Désormais, la variable « mon_nombre » possède le nombre **24** et non la chaîne de caractère "24".

Des opérations arithmétiques peuvent donc être effectuées sur la variable « mon_nombre ».

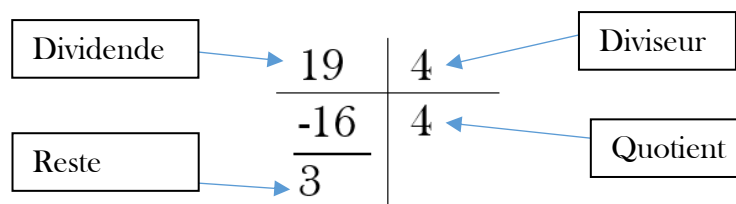
Attention, si la conversion échoue (ex : `int("abc")`), le programme plante !

E. Opération sur les nombres

Code Python	Résultat
<code>print(24 + 7)</code>	31
<code>print(10 - 2)</code>	8
<code>print(6 * 7)</code>	42
<code>print(15 / 5)</code>	3
<code>nb1 = 4</code> <code>nb2 = 8</code> <code>print(nb1 + nb2)</code>	12
<code>ze = "512"</code> <code>h = 22</code> <code>print(int(ze) + h)</code>	534
<code>print(3 + 2 * 3)</code>	9

Une opération très utile est également le modulo %

Le modulo concerne le reste d'une division, par exemple $19\%4$ (prononcé 19 modulo 4) correspond au reste de la division entière de 19 par 4 :



Donc le résultat modulo est : $19 \% 4 = 3$

F. Les commentaires

Les commentaires en Python sont des lignes de texte dans le code qui sont ignorées par l'interpréteur Python. Ils servent principalement à rendre le code plus compréhensible pour les humains. En effet, ils permettent d'expliquer le but des sections de code, des fonctions, des variables ou des algorithmes complexes. Il assure la maintenance du programme afin que d'autres développeurs (ou vous-même à l'avenir) travaillant dessus puissent modifier et corriger le code. Des commentaires peuvent être extraits pour générer de la documentation.

Commentaire sur une seule ligne (#)

Le caractère `#` est utilisé pour commencer un commentaire. Tout ce qui suit `#` sur la même ligne est considéré comme un commentaire.

Exemple :

```
#ceci est un commentaire sur une seule ligne
x = 10 # Commentaire en fin de ligne expliquant la variable x
```

Commentaire sur plusieurs lignes (""" """)

Les commentaires sur plusieurs lignes s'effectuent avec trois guillemets.

```
""" Ceci est un
commentaire sur plusieurs
lignes """
''' Ceci est également
un commentaire
sur plusieurs lignes '''
```

Quand et comment commenter ?

Au lieu de décrire ce que le code fait (qui est souvent évident en lisant le code lui-même), expliquez pourquoi il le fait et quel est son objectif. Ne commentez pas ce qui est déjà clair. Les commentaires doivent être faciles à lire et à comprendre. Par exemple, le commentaire suivant est inutile :

```
x = 7 # x vaut 7
```

Bon commentaire :

```
# Calcul de la consommation électrique
```

G. Erreurs fréquentes

⇒ Oublier les guillemets " " :

```
print("Bonjour tout le monde)
```

```
SyntaxError: EOL while scanning string literal
```

⇒ Confondre texte et nombre

```
text = "2390"
```

```
a = 10 + text
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

⇒ Oublier `int()`

```
age = input("Entrer votre age")
```

```
print(f"Dans un an, tu auras {age+1} ans")
```

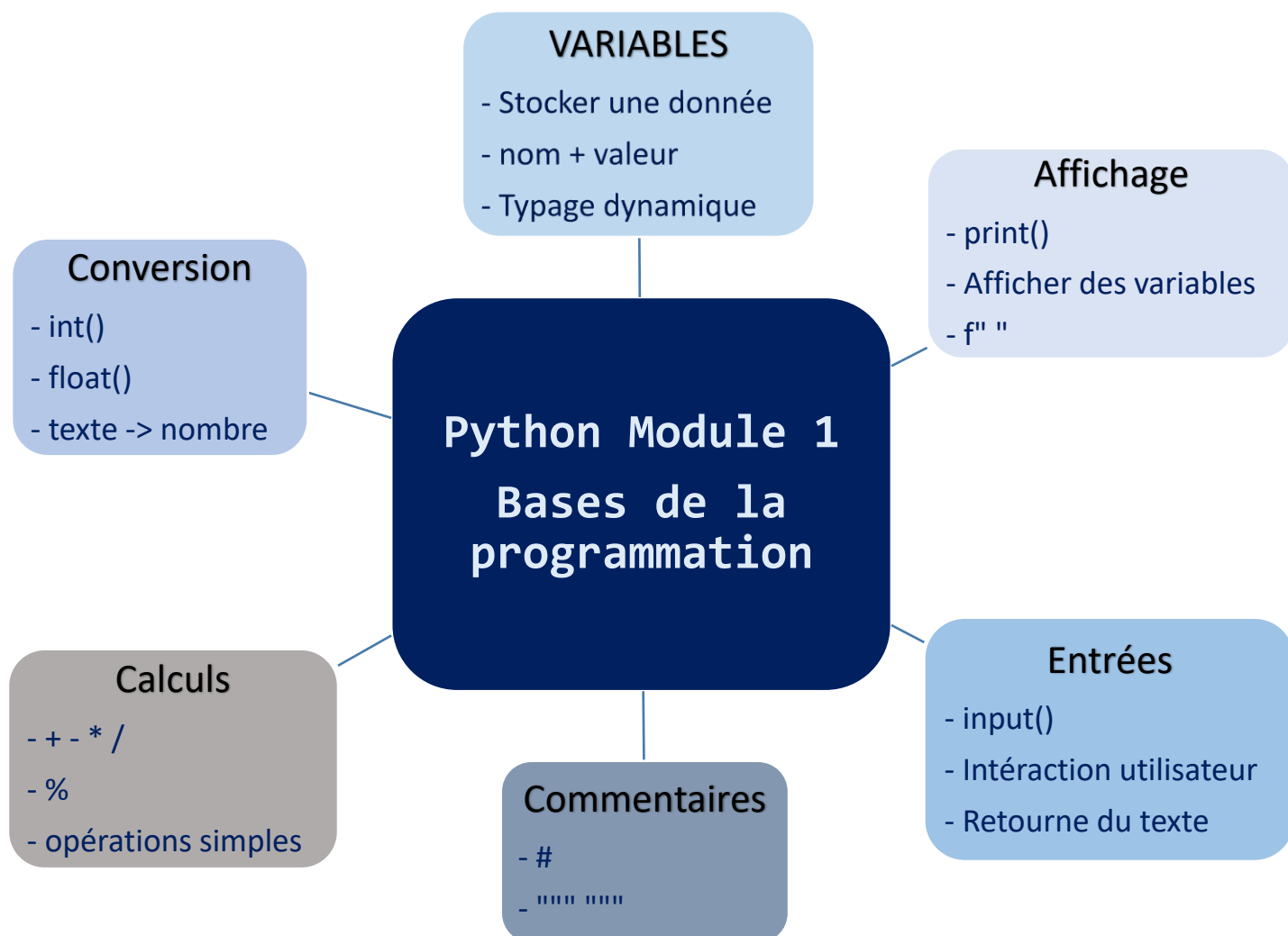
```
TypeError: can only concatenate str (not "int") to str
```

⇒ Oublier les parenthèses de `print()`

```
print "Coucou !"
```

```
SyntaxError: Missing parentheses in call to 'print'.
```

H. Synthèse



POINTS CLÉS A RETENIR

- input() retourne toujours du texte
- int() permet de convertir en nombre
- print() permet d'afficher
- une variable stocke une donnée