

Les expressions

Notion d'expression

Une **expression** est une portion de code que l'interpréteur Python peut évaluer pour obtenir une valeur.

Exemple :

```
>>> 2*5+6
>>> 7/4
>>> 7//4
>>> 'Attention' + 'aux espaces'
>>> input('Entrez une valeur')
>>> 3 in 1,2,3
```

En mode console, Python affiche par défaut la valeur obtenue. Mais en mode script, le programme n'affichera une valeur que si vous le demandez **explicitement** avec la fonction print.

Syntaxe Python

instruction sur plusieurs lignes terminer chaque ligne par un antislash \

plusieurs instructions sur une ligne séparer les instructions avec des points-virgules

Commentaire : Le symbole # permet de faire figurer dans le corps du programme un commentaire qui ne sera pas pris en compte lors de son exécution.

print pour afficher les valeurs.

- syntaxe : `print(value, ..., sep=' ', end='\n')`
`sep` est une chaîne qui sépare les différentes valeurs, par défaut un blanc.
`end` est une chaîne placée après la dernière valeur, à défaut un retour à la ligne (`\n`)

input arrête le programme et permet à l'utilisateur de donner une valeur.

- syntaxe : `input([prompt]) -> string`. On écrira donc `Var=input([prompt])`.
- La chaîne `prompt` est affiché avant la lecture de la valeur (cela permet de poser une question ou de placer un commentaire).
- Attention cette valeur est systématiquement une chaîne de caractère, on utilise alors la syntaxe
`Var=int(Var) ; Var= float(Var) ; Var=complex(Var)`
pour affecter la bonne valeur à la variable `Var`.