Haute École Bruxelles-Brabant
École Supérieure d'Informatique
Bachelor en Informatique

2023 – 2024
DEV1 – DEV1L

DEV1 – DEV1L – Laboratoires Python

TD 02 – Variables et Expressions

Nous poursuivons notre apprentissage du langage Python en introduisant la notion de variable (permettant de stocker des données à un endroit de la mémoire) et celle d'expression. Nous apprendrons également à lancer Python en mode script et à écrire nos premiers programmes.

Table des matières

Version du document
17 septembre 2023

Avertissement

Les laboratoires de DEV1 sont progressifs : vous devez absolument avoir fait et compris le TD01 *avant* d'aborder celui-ci. Si ce n'est pas le cas, faites d'abord le TD01 et ne revenez à ce TD-ci qu'une fois le TD01 complètement terminé. Votre professeur de labo peut vous aider si vous n'avez pas compris certaines notions introduites dans le TD01 : *n'hésitez pas à lui demander de l'aide* le cas échéant !

1 Les variables en Python

En Python, une *variable* est un nom symbolique permettant de faire référence à une valeur stockée en mémoire. Les commandes suivantes déclarent trois variables référençant un nombre, une chaîne de caractère, et une liste (respectivement) :

```
nombre_jours = 365
```

```
message = 'Bienvenue à tous'
```

```
notes_etudiants = [10,15,12,18,5,0,15]
```

```
>>> nombre_jours = 365
>>> type(nombre_jours)
<class 'int'>
>>> message = 'Bienvenue à tous'
>>> notes_etudiants =
[10,15,12,18,5,0,15]
```

Pour connaître le type d'objet référencé par une variable, on utilise la commande : `type(nom_variable)`.

```
>>> type(message)
<class 'str'>
>>> type(notes_etudiants)
<class 'list'>
```

Exercice 1 **Les variables en Python**

Lancer Python en mode interactif et exécutez les commandes permettant d'arriver aux résultats demandés, inscrivez ensuite la commande que vous avez utilisé dans le cadre :

1. Déclarer une variable `age_client` référençant le nombre `21` : `>>> age_client=21`
2. Déclarer une variable `nom_client` référençant la chaîne `'Charlot'` : `>>> nom_client='Charlot'`
3. Déclarer une variable `temperature` référençant le nombre `17.5` : `>>> temperature=17.5`
4. Afficher le type de donnée référencé par `age_client` : `>>> type(age_client)` `<class 'int'>`
5. Changer la donnée référencée par `age_client` qui référence à présent une liste contenant les nombres 1,2 et 3 (dans cet ordre) : `>>> age_client=[1,2,3]`
6. Afficher à nouveau le type de donnée référencé par `age_client` : `>>> type(age_client)` `<class 'list'>`

Instances et références en Python

Les données en Python sont vues comme des *objets* (aussi appelés *instances*) et sont stockés dans la mémoire. Les variables sont des *références* vers ces objets. Comme nous avons pu le voir dans l'exercice ci-dessus, l'objet référencé par une variable peut changer à tout moment. En particulier, contrairement à d'autres langages de programmation (comme Java par exemple), on ne doit pas préciser le type d'une variable. On peut par contre le connaître grâce à la fonction `type`.

2 Les expressions

Avec les variables, les littéraux (valeurs apparaissant telles quelles dans une commande) les opérateurs et les fonctions mises à notre disposition par Python (voir TD01) nous pouvons former des *expressions* de façon à dériver d'autres valeurs .

Exercice 2 Expressions

Déclarez les quatre variables suivantes :

1. `lageur` référençant le nombre `2.8`,
2. `longueur` référençant le nombre `34`,
3. `texte` référençant la chaîne `'le langage Python'`,
4. `ma_liste` référençant une liste contenant les chaines `'A'`, `'B'` et `'C'` (dans cet ordre).

Demandez alors à Python d'afficher la valeur des expressions suivantes. Certaines d'entre-elles sont erronées, veuillez en donner la raison (dans la colonne de droite). Pour celles qui ne le sont pas, donnez la valeur de l'expression.

Expression	Raison de l'erreur ou valeur de l'expression
`longueur * largeur`	95.19999999999999
`largeur**3`	21.951999999999995
`2*longueur + 2*largeur`	73.6
`(longueur + largeur)/2`	18.4
`longueur + largeur/2`	35.4
`longueur + texte`	TypeError: unsupported operand type(s) for +: 'int' and 'str'
`longueur - len(texte)`	17
`max(longueur,largeur,len(texte))`	34
`ma_liste*2`	['A', 'B', 'C', 'A', 'B', 'C']
`ma_liste+2`	TypeError: can only concatenate list (not "int") to list
`texte[0:10]`	'le langage'
`texte[0]*10`	'llllllllll'
`largeur//10`	0.0
`texte//10`	TypeError: unsupported operand type(s) for //: 'str' and 'int'

2,8=0*10+2,8

Toute expression valide peut être stockée en mémoire et référencée par une variable. On utilise pour cela l'opérateur `=`, aussi appelé *opérateur d'assignation* :

```
ma_variable = expression
```

Exercice 3 Assigner une expression à une variable

En utilisant les variables de l'exercices précédent, exécutez les commandes suivantes. Attention : certains d'entre-elles sont erronées. Inscrivez la valeur de la variable impliquée *après* exécution de la commande dans la colonne de droite, ou la raison pour laquelle l'assignation est erronée sinon.

Assignation	Valeur de la variable (ou cause de l'erreur)
`longueur = (5 + 3**10)/3`	19684.666666666668
`longueur = largeur`	longueur vaut donc 2.8
`5 = longueur`	SyntaxError: cannot assign to literal here
`longueur = longueur + 5`	7.8
`largeur = largeur/2 - 1`	0.3999999999999999
`longueur + largeur = texte`	SyntaxError: cannot assign to expression here
`texte = texte*5`	le langage Pythonle langage Pythonle langage Pythonle langage etc
`longueur = texte`	le langage Pythonle langage Pythonle langage Pythonle langage etc
`longueur = longueur +1`	TypeError: can only concatenate str (not "int") to str

3 Python en mode script

Jusqu'à présent, nous avons travaillé avec Python en mode interactif. Nous sommes cependant prêts à écrire des petits scripts : une suite de commandes Python, contenues dans un fichier, et que Python sera chargé d'exécuter.

Un tel script sera la plupart du temps amené à interagir avec l'utilisateur et, pour cela, nous aurons besoin d'utiliser deux fonctions particulières : `print` et `input`.

Exercice 4 Les fonctions print() et input()

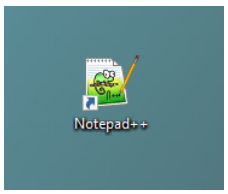
Utiliser la fonction `help()` de Python pour obtenir de l'aide sur les fonctions `print` et `input`. À quoi servent ces deux fonctions ?

Fonction	Utilité de la fonction ?
`print()`	Prints the values to a stream, or to sys.stdout by default
`input()`	Read a string from standard input.

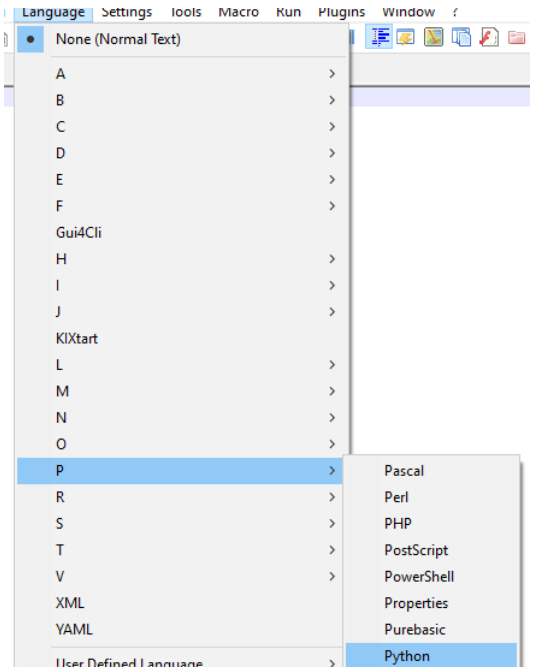
Le tutoriel suivant vous guide dans l'écriture de votre premier script.

Tutoriel 1 Écrire et exécuter un script Python

- Lancez l'éditeur de texte `Notepad++`, l'icône se trouve sur votre Bureau.



- Configurez l'éditeur pour coder en Python. Pour cela, allez dans le menu `Language` et cliquez sur `Python` :

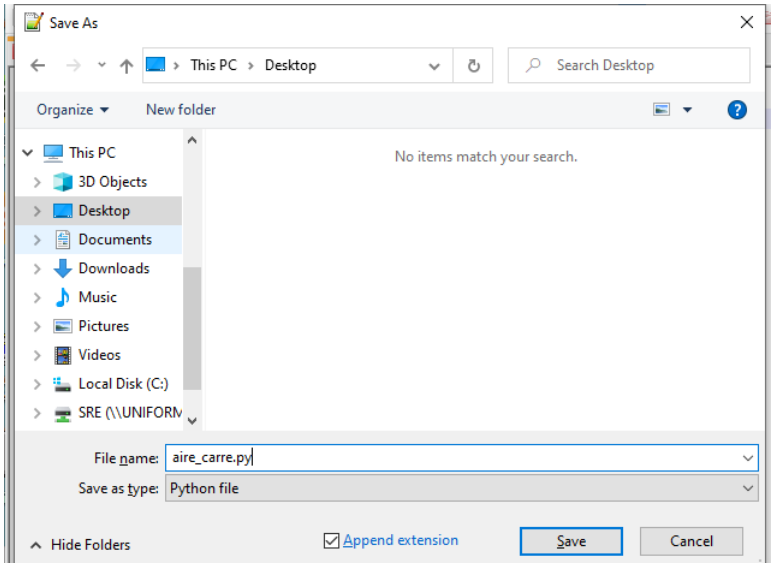


- Inscrivez ensuite le code suivant dans la fenêtre :

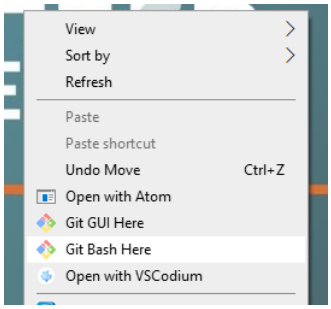
airecarre.py

```
1 # Mon premier script Python !
2 #
3 # Un programme calculant l'aire du carré dont la
4 # longueur du côté est entrée au clavier par l'utilisateur
5
6 print(" **** Bienvenue ! **** ")
7 x = input("Entrez la valeur du côté du carré : ")
8 x = int(x)
9 print("L'aire du carré vaut:", x**2)
```

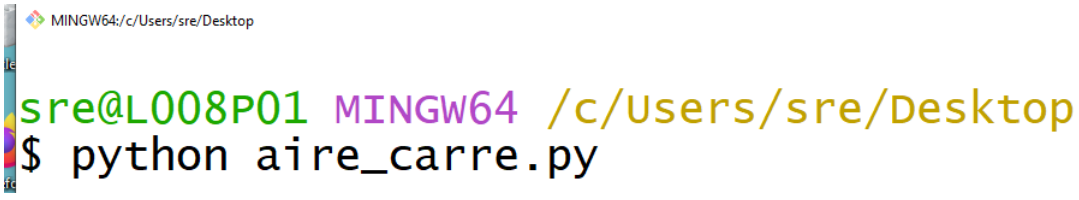
Sauvez le fichier sur le bureau en cliquant sur `save`, appelez le fichier `aire_carre.py`



Cliquez droit sur le bureau et choisissez `git bash here` :



Demandez à Python d'exécuter le fichier en tapant `python aire_carre.py` :



Vous constatez alors l'exécution du script.

Expliquons le fonctionnement du code ci-dessus :

- Lors de l'exécution du script, Python exécute toutes les commandes du fichier l'une à la suite de l'autre, en commençant par la première commande.
- Les lignes 1 à 4 contiennent des *commentaires* : ces lignes sont ignorées par Python.
- À la ligne 6 la fonction `print` est utilisée pour afficher une message de bienvenue à l'utilisateur.

- ▷ À la ligne 7, la fonction `input` est utilisée pour demander un nombre à l'utilisateur. Le résultat de la lecture est alors stocké sous la forme d'une chaîne de caractère dans la variable `x`.
- ▷ Comme la fonction `input` renvoie toujours une chaîne de caractère, il est nécessaire de convertir cette chaîne en nombre pour pouvoir faire nos calculs : c'est le but de la ligne 8.
- ▷ Finalement, la ligne 9 affiche la valeur de l'aire (qui est simplement x^2), précédé d'un message explicatif.

Les lignes 7 et 8 du script auraient pu être écrites plus simplement :

```
x = int(input("Entrez la valeur du côté du carré : "))
```

De même que la fonction `int()`, la fonction `float()` permet de convertir une chaîne de caractère en nombre à virgule. La commande `float(x)` déclenchera une erreur si `x` contient autre chose que des chiffres un et `.` servant de virgule.

4 Vos premiers programmes

Vous allez ici vous exercez en douceur à l'écriture de scripts Python en modifiant l'exemple donné à la section précédente.

Exercice 5 Le volume d'un cube

Modifiez le script `aire_carre.py` de façon à écrire un script permettant de calculer le volume [1] d'un cube de côté `x`, où `x` est un nombre lu au clavier. Pour ce faire, nous vous demandons de suivre les étapes suivantes :

le prochain labo

1. Créez un nouveau fichier `XXX.py` ou vous remplacerez `XXX` par un nom évoquant la fonction du script,
2. Copiez-collez le code du fichier `aire_carre.py`,
3. Adaptez les commentaires du début du fichier (commenter correctement son code est important),
4. Adaptez l'appel à la fonction `input` de façon à demander la longueur de l'arête d'un cube à l'utilisateur,
5. Adaptez le `print` final.

Exercice 6 L'aire et le périmètre d'un cercle

Même question qu'à l'exercice précédant, mais on souhaiterait cette fois disposer d'un script permettant de calculer à la fois l'aire et le périmètre [2] d'un cercle de rayon `r` lu au clavier.

Le script doit afficher l'aire du cercle, passer à la ligne, afficher le périmètre du cercle. A vous de voir comment adapter correctement le script `aire_carre.py` pour arriver à ce résultat !

Exercice 7 L'aire d'un rectangle

1. Rappel : le volume V d'un cube de côté x est $V = x^3$
2. Rappel : l'aire A d'un cercle de rayon r est donnée par $A = \pi r^2$ tandis que le périmètre P est donné par $P = 2\pi r$, vous pouvez utiliser l'approximation $\pi \approx 3.14$

Même question qu'à l'exercice précédant, mais on souhaiterait cette fois disposer d'un script permettant de calculer l'aire[3] d'un rectangle de longueur l et de largeur L, où l et L sont lus au claviers.

Exercice 8 — Temps en secondes

Écrivez un script qui demande un nombre d'heures, un nombre de minutes et un nombre de secondes et qui affiche le nombre de secondes totales.

Par exemple : si l'utilisateur entre 2 heures, 10 minutes et 27 secondes, le programme affiche 7827. En effet 2 heures donnent 7200 secondes, 10 minutes sont 600 secondes auxquelles il faut ajouter les 27 secondes : 7200 + 600 + 27 = 7827.

5 En résumé ...

Principaux points de matière du TD

Voici les principaux points abordés lors de ce TD. Vous devez absolument être à l'aise avec ceux-ci avant d'aborder la prochaine séance d'exercice.

1. Déclarer et manipuler des variables en Python, utiliser la fonction `type()` pour connaître leur type.
2. Écrire et manipuler des expressions.
3. Écrire et exécuter un script Python, utiliser les fonctions `print()` et `input()` pour afficher des données à l'écran et lire des données au clavier.

3. Rappel : l'aire d'un rectangle de longueur l et de largeur L est donné par $A = l.L$.