

.....

Session d'exercices – Les opérateurs et expressions

Opérateurs arithmétiques et logiques

L'objectif de cet exercice est d'utiliser les opérateurs arithmétiques et logiques et d'afficher à l'écran le résultat de vos calculs.

Pour commencer ce TP, il faut d'abord lancer l'interpréteur en tapant la commande "python" (ou python3) dans le terminal (pas besoin de Visual Studio Code cette semaine, donc):

```
user@epfl:$ python
```

Dans cet exemple "user@epfl:\$" est le message que devrait déjà afficher votre terminal. Il représente votre utilisateur qui est connecté à la machine virtuelle. Si vous utilisez votre propre machine cela affichera probablement quelque chose de différent, mais le principe reste le même. Comme nous l'avons vu plus tôt dans le semestre, l'interpréteur nous permet d'entrer des expressions python ligne par ligne : à chaque ligne, l'interpréteur nous affiche la valeur à laquelle l'expression entrée a été évaluée. Si vous tapez donc `2 + 3` dans l'interpréteur, il vous répondra `'5'`, et si vous entrez le nom d'une variable, il vous affichera la valeur qu'elle contient. L'interpréteur est aussi capable de se souvenir de ce qui a été fait plus tôt dans la session. Il est donc possible de créer des variables qu'on pourra utiliser ensuite, comme dans un script python classique.

Chaque nouvelle ligne de l'interpréteur commence par `'>>>'` pour vous indiquer que c'est à vous d'écrire. Dans ce TP, nous allons tout faire à travers l'interpréteur. Pour quitter l'interpréteur, entrez l'instruction `'exit()'` et tapez «ENTER», ou appuyez sur `Ctrl + D`.

Types, initialisation et affectation des variables

Comme vu pendant le cours, chaque variable a un type et une valeur. Lors de l'initialisation de la variable, le type est déduit automatiquement de la valeur d'initialisation.

Exercice 1 [Difficulté: *]

1. Dans l'interpréteur, exécutez les commandes suivantes:

```
i1 = 0
pi = 0.0
b = True
```

.....

2. Affichez les valeurs et les types des variables `i1`, `pi` et `b`. Expliquez pourquoi elles ont le type qui est affiché. Rappelez-vous de la fonction `"type()"`.
3. Changez la valeur de `i1` à 55.
4. Initialisez des nouvelles variables : `i2` à 22, `i3` à 42 et `m` à 77.
5. Changez la valeur de `pi` à 3.14.
6. Affectez la valeur `False` à `b`.

Opérateurs arithmétiques: `+`, `-`, `+=`, `-=`

Exercice 2 [Difficulté: *] Effectuez les changements suivants sur `i4` et affichez la nouvelle valeur de `i4` après chaque changement :

1. Affectez la valeur de `i1` à `i4`.
2. Affectez la somme des variables `i2` et `i3` à `i4`.
3. Affectez le résultat de la soustraction de `i3` et `i1` à `i4`.
4. Incrémentez `i4` de 6.
5. Décrémentez `i4` de 3.

Opérateurs arithmétiques: `/`, `//`, `%`, `*`, `**`

Exercice 3 [Difficulté: *] Exécutez les commandes suivantes et affichez la valeur et le type de `i4` après chaque modification :

```
i4 = i3 / i2
i4 = i3 // 27
i4 = i3 % 4
i4 = i1 ** 2 * i2
```

Expliquez pourquoi `i4` a eu les types affichés.

Application pratique du Python

Python a de nombreuses applications pratiques. Il est utilisé par des informaticiens pour l'apprentissage des réseaux de neurones artificiels, par les mathématiciens pour les calculs numériques, par les biologistes pour le séquençage des gènes et par vous pendant ce semestre (et avec espoir aussi à l'avenir !). Voici quelques simples exemples d'applications pratiques de Python.

.....

Exercice 4 [Difficulté: *] Calculez $i_4 = 5i_2^2 + 17i_2 + 88$ et affichez le résultat (variable `i4`).

Exercice 5 [Difficulté: *] Si le radius d'un cercle est $r = 1.1$ mètres, calculez sa surface. Utilisez la variable `pi` définie précédemment.

Exercice 6 [Difficulté: *] Un cône de révolution est dans un cylindre régulier. Ils partagent la même base dont le radius est $r = 2.11$ mètres et ils ont la même hauteur `h` de 5 mètres. Calculez la partie du volume du cylindre en dehors du cône. (La volume d'un cône est $V_{\text{cône}} = \frac{\pi r^2 h}{3}$).

Exercice 7 [Difficulté: *] Si la densité de l'hélium est $0.179 \frac{\text{kg}}{\text{m}^3}$, calculez le poids de 40 m^3 d'hélium.

.....

Opérateurs booléens: **and**, **or** et **not**

Souvenez-vous du cours que les opérateurs **and** et **or** sont « paresseux », c'est à dire que python ne va pas évaluer l'argument à droite de **and** si celui à gauche vaut **False**, et idem pour **or** si l'argument à gauche vaut **True**.

Exercice 8 [Difficulté: *] Initialisez les variables suivantes :

```

pizza = True
drive = True
cocoa_butter = True
pineapple = True
sugar = True
drink = True
milk = True
bus = False
train = True
cocoa = False

```

Exécutez le code suivant et affichez les valeurs des nouvelles variables après chaque commande (où **b** et **m** ont les valeurs que vous leur avez données dans les exercices précédents) :

```

jail = drink and drive
sin = pizza and pineapple
milk_chocolate = cocoa and cocoa_butter and sugar and milk
white_chocolate = not(cocoa) and cocoa_butter and sugar and milk
transport = bus or train
n1 = 34 - (m or (not b))
n2 = 34 - ((not b) or m)

```

Essayez de comprendre des nouvelles valeurs. Pourquoi la valeur de la variable **n1** est-elle différente de celle de **n2** ? Assurez-vous d'avoir bien compris.

Comparaisons et expressions logiques

Rappelez-vous ce que vous avez vu en cours : dans le contexte d'une expression booléenne, les différentes valeurs n'ont pas forcément besoin d'être des booléens (**True** et **False**). Par exemple, dans le contexte d'une expression booléenne, un nombre sera interprété comme **True** s'il ne vaut pas zéro, et une chaîne de caractères sera interprétée comme **True** si elle n'est pas vide.

.....

Exercice 9 [Difficulté: *] Exécutez les lignes de code suivantes (où `i1`, `i2`, `i3` et `i4` ont les valeurs que vous leur avez données dans les exercices précédents), affichez les résultats après chaque ligne et essayez de bien les comprendre :

```
b4 = (i1 < i2) and (i3 > i1)
b4 = i1 or (i3 and (i2 <= i1))
b4 = (i1 == i2) or (i2 != i3)
b4 = 15 < i4 < 89
```

Exercice 10 [Difficulté: *]

1. Écrivez une expression qui sera vraie (`True`) si `i2` et `i3` sont strictement positives.
2. Écrivez une expression qui sera fausse (`False`) si `i2` est égale au reste de la division de `i3` par 7.
3. Écrivez une expression qui sera fausse si `i2` n'est pas égale à `i3`.

Exercice 11 [Difficulté: *] Admettons que la tranche d'âge pour pouvoir monter sur une montagne russe dans un parc d'attraction est entre 10 et 60 ans (inclus), et que Thomas est âgé de 12 ans. Créez une variable `age` initialisée à 12, et écrivez une expression qui est égale à `True` si Thomas peut monter sur l'attraction, et `False` sinon.

Exercice 12 [Difficulté: **] En vous inspirant de votre solution à l'exercice précédent, écrivez une expression qui est égale à "yes" si Thomas a le droit de monter sur la montagne russe, et "no" sinon. Il faut donc faire quelques ajouts pour que l'évaluation ne s'arrête pas à `True` ou `False`, mais qu'elle continue pour évaluer la bonne chaîne de caractères. Essayez de changer la valeur de la variable `age` et de relancer l'évaluation de votre expression pour voir si elle fonctionne correctement. (Rappelez-vous que vous pouvez utiliser les flèches haut et bas pour parcourir l'historique des commandes que vous avez tapées dans un terminal ou un interpréteur python).

Exercice 13 [Difficulté: **] Écrivez une expression qui affiche la chaîne de caractères "pair" si `i4` est pair, et "impair" dans le cas contraire. Rappelez-vous que, comme dans les précédents exercices, on attend toujours une seule

.....

expression, c'est à dire une seule ligne qui évalue à la bonne chaîne de caractères en fonction de la valeur de `i4`. Essayez de relancer votre expression avec différentes valeurs de `i4` pour vous assurer qu'elle est correcte.

Exercice 14 [Difficulté: *]** Écrivez une *unique* expression qui est égale à :

- "Fizz" si `i4` est divisible par 3,
- "Buzz" si `i4` est divisible par 4,
- "FizzBuzz" si `i4` est divisible à la fois par 3 et par 4,
- la valeur de `i4` si `i4` n'est divisible ni par 3 ni par 4.

On attend bien sûr toujours une seule expression utilisant les opérateurs booléens, qui est évaluée à différentes valeurs en fonction de la valeur de `i4`, sans passer par des variables intermédiaires. On rappelle aussi qu'un nombre `x` est dit divisible par un nombre `y` si le reste de la division de `x` par `y` vaut zéro.