

Outils Info. TP1: Variables et Programmes Python

Important : rendu sur claroline

Vos réponses seront rendues dans une archive zip : cette archive contiendra un fichier `compte-rendu.txt` (pour les réponses aux questions précédées par une apostrophe) et vos fichiers pythons.

Pour faire une **archive zip** de votre TP pour le rendu, exécutez, dans le terminal :

```
cd ~/outils-info/  
zip -r tp1.zip tp1/
```

Important : mise en place

Comme pour tous les TPs :

- travaillez dans un dossier dédié au TP, lui même dans `~/outils-info`,
- créez un fichier `compte-rendu.txt` pour écrire la date, vos noms et les réponses aux questions précédées d'une apostrophe,
- créez un sous-dossier par exercice (sauf instructions contraires).

Pour démarrer, vous pouvez par exemple exécuter les commandes :

```
cd ~/outils-info/  
mkdir tp1  
cd tp1  
emacs compte-rendu.txt &
```

Exercice 1 – Le premier programme python

Pour cet exercice, travaillez dans un dossier `ex1` lui même dans le dossier `tp1`.

'Q1) Créez et ouvrez un fichier `statique.py` pour écrire le programme des questions à venir. Quelle(s) commande(s) avez-vous utilisée(s) ?

Q2) Définissez deux variables : `p` initialisée avec la valeur 5 et `q` initialisée avec la valeur 3p.

Q3) Faites que le programme affiche « `p vaut 5 et q vaut 15, leur somme vaut 20` ».

'Q4) Testez votre programme. Quelle commande avez-vous lancé ?

'Q5) Copiez votre programme et appelez la copie `avecinput.py` et ouvrez la copie pour écrire le programme des questions à venir. Quelle(s) commande(s) avez-vous utilisé ?

Q6) Modifiez votre programme pour qu'il demande successivement les valeurs des entiers `p` et `q` à l'utilisateur.

'Q7) Testez votre programme avec différentes valeurs.

L'opérateur `%` calcule le reste de la division entière (appelé aussi « modulo »), par exemple `121 % 50` (« 121 modulo 50 ») vaut `21`.

'Q8) Comment formuler « q est-il un multiple de p » à l'aide du modulo ?

Q9) Traduisez en Python (et complétez votre programme) « si q est un multiple de p, afficher `Multiple` sinon afficher `Pas multiple` ». Testez votre programme.

Q10) Complétez votre programme avec des lignes de codes qui permettent d'échanger les valeurs des variables `p` et `q`, en utilisant une nouvelle variable que vous appellerez `tmp` (temporaire).

Q11) Faites que votre programme affiche à nouveau les valeurs de `p` et `q` et leur somme. Vérifiez que votre programme fonctionne.

Q12) Challenge optionnel : échangez à nouveau les valeurs de `p` et `q`, mais sans utiliser d'autres variables. Soyez astucieux, jouez avec des opérations arithmétiques...

Exercice 2 – Les deux plus grands

Pour cet exercice, travaillez dans un fichier `ex2.py`, directement dans le dossier `tp1`.

Q13) Écrivez un programme demandant successivement trois entiers à l'utilisateur (que l'on pourra appeler `a`, `b`, `c`) et affichant la valeur de la somme des deux plus grands. Astuce : pour afficher la somme des deux plus grands, il est utile de trouver d'abord lequel est le plus petit des trois nombres.

Q14) Testez votre programme avec plusieurs valeurs pour les paramètres. Quelles commandes avez-vous lancées pour savoir que votre programme marche ?

Exercice 3 – Le jeu des N questions

Pour cet exercice, travaillez dans un fichier `ex3.py`, directement dans le dossier `tp1`.

Le but est d'écrire un programme qui devine à quel animal vous êtes en train de penser en vous posant des questions auxquelles vous ne pouvez répondre que par « oui » ou par « non » (si vous répondez autre chose, le programme doit afficher un message d'erreur et arrêter le jeu).

Pour simplifier le problème, on se limitera à une liste fermée de 6 animaux : oiseau, moustique, chat, chien, tyrannosaure et ptérodactyle.

Q15) Imaginez des questions fermées (réponse oui/non) pouvant permettre de trouver l'animal que l'utilisateur a choisi parmi la liste donnée. Combien de questions devez vous avoir, au minimum, pour être sûr de pouvoir trouver n'importe quel animal parmi les 6 possible ?

Q16) Écrivez un programme qui pose ces questions à l'utilisateur pour finalement afficher à quel animal il pense. Attention à ne pas embêter l'utilisateur et donc à ne poser que les questions qui sont utiles.

Exercice 5 – Racines de polynômes

Pour cet exercice, travaillez dans un fichier `ex5.py`, directement dans le dossier `tp1`.

Q18) Écrivez un programme qui calcule et affiche les racines d'un polynôme du second degré $ax^2 + bx + c$. Les valeurs de `a`, `b` et `c` sont des nombres réels que le programme demande à l'utilisateur. Pensez à bien gérer les différents cas.