

Programmation I - TP 1

1 Compréhension

1.1 Affectations

On exécute les instructions suivantes :

```
tab = [4, 8, 15, 15, 23, 42]
x = tab[0]
tab[3] = 16
x = x + 3
y = x + 2
```

Quelle sont les valeurs de `x`, `y` et `tab` après exécution de ces instructions ?

1.2 Appels de fonction

```
def fonc (x):
    if x < 0:
        x = -x
    while x > 1:
        x = x - 2
    if x == 0:
        return True
    else:
        return False
```

Quelle serait la valeur de retour de `fonc(3)` ? `fonc(0)` ?

D'une manière générale, quand est-ce que `fonc` retourne `True` ?

A quoi cette fonction peut-elle servir ?

1.3 Priorité des opérateurs

On exécute les instructions suivantes :

```
x = ((2 * 5) + 4)
y = ((3 * (2 + 1)) == 9)
vv = (False or (True and ((2 * 2) == 4)))
```

Combien y-a-t-il d'opérateurs sur chaque ligne ?

Quelles sont les valeurs de `x`, `y` et `vv` ?

Toutes les parenthèses ci-dessus ne sont pas utiles¹. Y'a-t-il des parenthèses utiles *parmi les deux premières lignes* ? Si oui lesquelles ?

¹Une paire de parenthèses est utile si le fait de la retirer modifie l'ordre dans lequel les opérations sont effectuées.

2 Complétion

2.1 Indentation

Ajouter des espaces pour que le code suivant soit correctement indenté :

```
esp = " "  
def compter_mots (phrase):  
    nb_mots = 1  
    for caractere in phrase :  
        if caractere == esp :  
            nb_mots += 1  
    return nb_mots
```

2.2 Comptage

Ajouter une instruction de boucle pour que la fonction `longueur` retourne le nombre de caractères dans le mot passé en argument :

```
def longueur (mot):  
    cpt = 0  
    [...] :  
        cpt = cpt + 1  
    return cpt
```

2.3 Caractère interdit

Ajouter des instructions dans la boucle pour que la fonction retourne `False` si le caractère interdit (2ème argument) est présent dans le mot (1er argument) :

```
def mot_correct (mot, c_interdit) :  
    for car in mot :  
        [...] :  
            return True
```

3 Implémentation

3.1 Longueur moyenne

On fournit le code suivant :

```
def moyenne (valeurs):  
    somme = 0.0  
    for v in valeurs:  
        somme += v  
    return somme / len(valeurs)  
  
non_mots = ["_", ".", ",", ";", "?", "!", ""]
```

Ecrivez une fonction `longueur_moyenne` qui calcule la longueur moyenne des mots d'une phrase. Vous pouvez vous servir de la fonction `moyenne` fournie (qui calcule la valeur moyenne d'une liste de nombres) et de la liste `non_mots` des caractères n'appartenant pas à un mot dans une phrase :

```
def longueur_moyenne (phrase):  
    [...]
```

3.2 Mots interdits

Ecrivez une fonction `phrase_interdite` qui prend en paramètre une phrase et une liste de mots tabous, et retourne `True` si la phrase contient un des mots tabous, et `False` dans le cas contraire.

Indice : la fonction est bien plus courte (et plus facile à écrire) en utilisant l'opérateur `in` plutôt qu'avec des indices et/ou des slices. Vous êtes cependant libres de choisir comment votre fonction doit s'y prendre.

Vous pouvez tester votre fonction avec les arguments suivants :

```
phrase_interdite(" Ceci est une phrase de test .", [" test", " essai"])  
phrase_interdite(" Ceci est une phrase de test .", [" fleur", " arbre"])  
phrase_interdite(" Ceci est une phrase de test .", [" un", " le"])  
phrase_interdite(" Ceci est une phrase de test .", [" ceci", " cela"])
```

Votre fonction produit-elle le résultat attendu dans les deux derniers cas ? Si non, pouvez-vous expliquer pourquoi ?

3.2.1 Bonus

- Modifiez votre fonction `phrase_interdite` pour qu'elle fonctionne comme prévu dans les deux dernier cas d'utilisation. ²
- Essayez d'envisager des solutions pour gérer le cas d'un mot tabou au pluriel.

²Vous pouvez vous aider de la fonction présentée à cette adresse : http://www.tutorialspoint.com/python/string_lower.htm. Attention, cette fonction ne marche pas exactement comme celles dont on s'est servi jusque là !

Préparation

Utiliser la machine virtuelle :

- Importer avec VirtualBox la machine virtuelle Linux (`C:\Documents and settings\All users\Documents partagés\Arch.ovf`) et la démarrer.
- Se connecter en ligne de commande (login : "etu", mdp : "relai"), et démarrer l'interface graphique avec la commande `startx`.

Une fois connectés, vous pouvez démarrer un terminal et entrer la commande `python3.2` pour accéder à l'interpréteur Python. Celui-ci vous permet d'entrer des instructions et les exécute au fur et à mesure. Une fois Python démarré :

- La commande `import [fichier]` exécute le contenu du fichier spécifié.
- La séquence `[Ctrl] + [C]` annule la commande en cours.
- La séquence `[Ctrl] + [D]` quitte l'interpréteur (comme la commande `exit()`).

Pour écrire un programme complet, vous pouvez utiliser l'éditeur de texte `geany` (enregistrez votre fichier avec l'extension `.py` pour bénéficier de la coloration syntaxique), puis importer votre fichier depuis l'interpréteur.

Le contenu de la machine virtuelle est perdu à chaque importation, et n'est pas transmis entre les postes du réseau de Bordeaux 3. Pour sauvegarder votre travail, vous pouvez récupérer vos fichiers sur clé USB, ou utiliser l'utilitaire `dropbox`, installé par défaut sur la machine virtuelle (nécessite un compte - gratuit).

Commandes utiles

Si vous utilisez le terminal Linux, voici quelques indications utiles :

- `pwd` - Indique le répertoire où vous vous trouvez.
- `ls` - Affiche le contenu du dossier courant.
- `cd [dir]` - Vous place dans le dossier `dir`, "`cd ..`" dans le dossier parent.
- Ajouter `&` après une commande l'exécute en arrière-plan. (Pour `geany`.)
- Comme dans l'interpréteur Python, `[Ctrl] + [C]` annule une commande.
- `cp src dst` - Copie le fichier `src` vers un fichier `dst`, `mv` le déplace.
- `[Tab]` déclenche l'auto-complétion. Le terminal essaye alors (s'il n'y a qu'une possibilité) de compléter la commande ou le nom de dossier/fichier en cours. S'il y en a plus d'une qui soit compatible avec les premiers caractères entrés, appuyer une deuxième fois affiche la liste des complétions possibles.
- Si vous utilisez Dropbox, le dossier spécial "`~/Dropbox/`" est automatiquement synchronisé avec votre espace de sauvegarde.