Séance 1 : Variables, fonctions et conditions

L1 – Université Côte d'Azur

Pour ce TP, nous vous demandons d'utiliser l'éditeur Thonny. On vous conseille fortement de l'installer sur votre ordinateur personnel pour travailler chez vous. Vous pouvez le télécharger à l'adresse : `https://thonny.org/`. Si vous tenez vraiment à utiliser un autre éditeur (GNU/Emacs, notepad++, Idle, etc.), ce sera à vous de vous assurer que tout marche correctement ! Nous vous conseillons cependant fortement de rester avec Thonny.

Création du répertoire de travail

Avant de commencer ce TP : sur votre ordinateur, créez un répertoire `Python` dans lequel vous rangerez tous vos fichiers de l'UE. Dans votre répertoire `Python` créer un répertoire `TP1`. Durant cette séance, tous vos fichiers devront être sauvegardés dans ce répertoire `TP1`.

Les prochaines semaines, vous devrez créer les répertoires `TP2`, puis `TP3`. Ainsi il sera facile de vous retrouver dans tous les fichiers.

Exercice 1 — Premiers pas dans la console (⋆)

1. Définissez deux variables : `p` ayant pour valeur 5 et `q` ayant pour valeur 3`p`.
2. Avec une seule instruction `print(…)` et en utilisant les variables `p` et `q`, faites afficher la phrase suivante : `p vaut 5 et q vaut 15 ; leur somme fait 20`.
   N.B. Votre code doit fonctionner même si `p` et `q` ont des valeurs différentes. Ainsi, la solution suivante n'est *pas* celle que l'on attend : `print('p vaut 5 et q vaut 15 ; leur somme fait 20')`.
3. Demander en python « `q` *est-il un multiple de* `p` ? ». Indication : utilisez le reste de la division euclidienne. Votre calcul doit renvoyer `True` ou `False`.
4. Sans consulter vos notes de cours, écrivez dans la console les lignes de code permettant d'échanger les valeurs de `p` et `q`, en utilisant une variable temporaire `tmp`. Vérifiez ensuite dans la fenêtre Variables [1] de Thonny que les valeurs de `p` et `q` ont bien été permutées.
5. Calculez la moyenne des entiers de 1 à 10. Le résultat est un nombre flottant (à virgule).
6. Calculez la moyenne des entiers de 1 à 11. Ne retapez pas la formule complète, mais modifiez la précédente en tapant sur la touche du haut [↑] qui va chercher les lignes précédemment entrées (l'*historique*). On peut utiliser [↓] en sens inverse.
7. Comment feriez-vous pour savoir si la fraction $\frac{51}{85}$ est irréductible ? En d'autres termes, peut-on la simplifier ? Par combien ?
8. Demandez en une ligne si 5^4 est plus grand que 4^5. Le résultat devra être un booléen.
9. Tapez dans la console : `help(max)` pour demander une petite documentation de la fonction `max`.

1. Si vous ne voyez pas de fenêtre variable, allez voir le cours 1 avec animation partie III « Espionner les variables avec Thonny ».

Exercice 2 — Utilisation de Thonny en mode debug (⋆)

1. Relisez le cours 1 (la version avec animation) partie IV les deux pages « Thonny : écrire des scripts » et « Thonny : raccourcis clavier ». Vous le trouverez sur ma page professionnelle :
   https://upinfo.univ-cotedazur.fr/~obaldellon/python
2. Créez un fichier `exercice2.py` dans votre dossier `TP1` que vous avez créé au début du TP. Ce fichier devra contenir le code suivant.

```
r = 0
m = int(input("Entrer un nombre entre 0 et 999 :"))

c = m%10
r = r*10 + c
m = m//10
print(c)

c = m%10
r = r*10 + c
m = m//10
print(c)

c = m%10
r = r*10 + c
m = m//10

print(c)
print(r)
```

3. Exécutez-le en appuyant sur la touche F5. Que fait-il ?
4. Exécutez-le pas à pas en appuyant sur les touches ⇧+F5 puis en appuyant plusieurs fois sur la touche F7. Observez.
5. Exécutez-le pas à pas en appuyant sur les touches Ctrl+F5 (pour passer en mode debug) puis en appuyant plusieurs fois sur la touche F7. Observez. Quelle est la différence ?

Exercice 3 — Générer une facture (⋆– ⋆⋆)

Un client veut acheter `n` produits de prix `p`. Le prix final est donc `n*p`. Il faut cependant ajouter le prix de la livraison et une éventuelle réduction.

1. À partir de `n` = 10, on applique une réduction de 10% et à partir de 50, la réduction devient 20%. Écrire la fonction `réduction(n)` qui renvoie selon la valeur de `n` une des valeurs parmi : 0, 10 ou 20. **Écrire au moins cinq tests.**

```
def réduction(n):
    if n>=50:
        return 20
    elif n>=10:
        return 10
    else:
        return 0

assert réduction(100) == 20
assert réduction(50) == 20
assert réduction(49) == 10
assert réduction(10) == 10
assert réduction(9) == 0
```

2. Le prix de la livraison correspond à 10% du prix initial (avant la réduction). Cependant, le prix de la livraison obtenue ne peut pas être inférieur à 5€ ou supérieur à 100€. Par exemple si le calcul des 10% donne moins que 5€

on fixera le prix de la livraison à 5€. Écrire la fonction `prix_livraison(prix)` qui prend en argument le prix initial et renvoie le prix de livraison. **Écrire au moins cinq tests**.

```python
def prix_livraison(prix):
    prix_l = prix*0.1 # 10% de prix
    if prix_l > 100:
        return 100
    elif prix_l < 5:
        return 5
    else:
        return prix_l

assert prix_livraison(200) == 20
assert prix_livraison(1000) == 100
assert prix_livraison(1001) == 100
assert prix_livraison(50) == 5
assert prix_livraison(49.5) == 5
```

3. Écrivez un programme `facture(n,p)` qui calcule le prix final en affichant les détails du calcul comme ci-dessous.

```
>>> facture(10,12)
Prix : 120 €
Réduction : - 10 %
Coût livraison : 12.0 €
----------------------
Prix total (dont livraison) : 120.0 €
```

```python
def facture(n,p):
    prix = n*p
    r = réduction(n)
    l = prix_livraison(prix)
    prix_final = prix * (1-r/100)+l
    print("Prix :",prix,"€")
    print("Réduction : -",r,"%")
    print("Coût livrasion :",l,"€ ")
    print("----------------------")
    print("Prix total (dont livraison) :",prix_final,"€")
```

Exercice 4 — Pierre-Feuille-Ciseaux (⋆⋆)

Le jeu pierre-feuille-ciseaux (aussi appelé parfois *mourre* en Italie ou en Chine) est un jeu de mains où deux joueurs s'affrontent en choisissant simultanément chacun un objet parmi *pierre*, *feuille* ou *ciseaux*. Si les deux joueurs choisissent le même objet, il y a égalité, sinon la feuille l'emporte sur la pierre, qui l'emporte sur les ciseaux, qui l'emportent sur la feuille. Vous allez maintenant programmer un jeu de mourre pour affronter votre ordinateur !

1. Créez un fichier `exercice4.py` dans votre dossier `TP1`.
2. Écrivez un script qui affiche la phrase « `Quel est ton choix, humain ?` » puis qui demande à l'utilisateur de saisir un texte, et enfin affiche la phrase « `Humain, tu as choisi XXXX` », où `XXXX` est remplacé par le texte saisi par l'utilisateur juste avant. Par exemple :

```
Quel est ton choix, humain ? pierre
Humain, tu as choisi pierre
```

Vous utiliserez la fonction `input(msg)` qui affiche le message contenu dans le paramètre `msg` et renvoie ce que l'utilisateur a saisi au clavier à la suite de ce message.

```python
choix_humain = input('quel est ton choix, humain ? ')
print('Humain, tu as choisi', choix_humain)
```

3. Modifiez votre programme pour qu'il affiche la phrase « `Ok` » si l'utilisateur a bien saisi l'un des trois mots *pierre*, *feuille*, ou *ciseaux*, et sinon affiche la phrase « `Choix incorrect, je vais choisir pour toi…` ». Pour l'instant on affiche juste le message, le choix sera fait dans les questions suivantes. Par exemple, on aura :

```
Quel est ton choix, humain ? pierre
Ok
```

ou encore

```
Quel est ton choix, humain ? puits
Choix incorrect, je vais choisir pour toi…
```

```
if choix_humain=='pierre' or choix_humain=='feuille' or choix_humain=='ciseaux':
    print('ok')
else :
    print('choix incorrect, je choisis pour toi...')
```

4. Revenez au début du programme et définissez une fonction `choix_ordi()` qui renvoie un mot au hasard parmi les trois mots possibles (vous aurez besoin de la fonction `randint` du module `random` vue en cours sur le transparent du cours intitulé : « Aléatoire : générer des entiers au hasard »). Complétez votre programme pour que le message d'erreur précédent indique le choix qu'a fait l'ordinateur. On aura désormais :

```
Quel est ton choix, humain ? puits
Choix incorrect, je choisis pour toi : feuille
```

```
from random import randint

def choix_ordi():
    choix = randint(0,2)
    if choix == 0 :
        return 'pierre'
    elif choix == 1 :
        return 'feuille'
    else :
        return 'ciseaux'
```

```
if choix_humain=='pierre' or choix_humain=='feuille' or choix_humain=='ciseaux':
    print('ok')
else :
    choix_humain = choix_ordi()
    print('Choix incorrect, je choisis pour toi : ', choix_humain)
```

5. Complétez votre programme pour faire une partie complète : l'ordinateur fait un choix pour lui au début (mais le garde secret), puis l'utilisateur fait un choix, puis le programme affiche les deux choix et le gagnant éventuel. Par exemple :

```
Quel est ton choix, humain ? ciseaux
ok
Humain : ciseaux
Ordinateur : feuille
Vainqueur : humain
```

Avant de déterminer le gagnant, on affiche les différents choix des deux joueurs.

```
choix_ordinateur = choix_ordi()
print('Humain :',choix_humain)
print('Ordinateur :',choix_ordinateur)
```

Solution 1 : on énumère tous les cas. C'est long et pénible (9 cas), à éviter

```python
if choix_humain == 'pierre' and choix_ordinateur == 'pierre' :
    print('Vainqueur : égalite')
if choix_humain == 'pierre' and choix_ordinateur == 'feuille' :
    print('Vainqueur : ordi')
if choix_humain == 'pierre' and choix_ordinateur == 'ciseaux' :
    print('Vainqueur : humain')
# ...
# Et on continue comme ça pour les 6 autres cas. C'est fastidieux
# En info, il faut être paresseux et chercher à faire plus simple.
```

Solution 2 : on énumère tous les cas, mais cette fois-ci en regroupant les tests avec deux niveaux de `if`. C'est déjà beaucoup plus lisible, mais ça reste laborieux.

```python
if choix_humain == 'feuille' :
    if choix_ordinateur == 'pierre' :
        print('Vainqueur : humain')
    if choix_ordinateur == 'feuille' :
        print('Vainqueur : égalite')
    if choix_ordinateur == 'ciseaux' :
        print('Vainqueur : ordi')

if choix_humain == 'ciseaux' :
    ...
```

Solution 3 : on ne fait plus que trois cas : l'égalité, la victoire de l'humain et sa défaite. Remarquez que l'on peux créer des variables booléennes (c'est à dire `True` ou `False`) `cas_gagnant_1`, `cas_gagnant_2` ou `cas_gagnant_3` pour rendre le test plus lisible.

```python
# Quels sont les trois cas qui font gagner l'être humain ?
cas_gagnant_1 = (choix_humain == 'pierre'  and choix_ordinateur == 'ciseaux')
cas_gagnant_2 = (choix_humain == 'feuille' and choix_ordinateur == 'pierre')
cas_gagnant_3 = (choix_humain == 'ciseaux' and choix_ordinateur == 'feuille')

if choix_humain == choix_ordinateur :
    print('Vainqueur : égalite')
elif cas_gagnant_1 or cas_gagnant_2 or cas_gagnant_3:
    print('Vainqueur : humain')
else :
    print('Vainqueur : ordinateur')
```

Version Complète

```python
from random import randint

def choix_ordi():
    choix = randint(0,2)
    if choix == 0 :
        return 'pierre'
    elif choix == 1 :
        return 'feuille'
    else :
        return 'ciseaux'

choix_humain = input('quel est ton choix, humain ? ')
print('Humain, tu as choisi', choix_humain)

if choix_humain=='pierre' or choix_humain=='feuille' or choix_humain=='ciseaux':
    print('ok')
else :
    choix_humain = choix_ordi()
    print('Choix incorrect, je choisis pour toi : ', choix_humain)

choix_ordinateur = choix_ordi()
print('Humain :',choix_humain)
print('Ordinateur :',choix_ordinateur)

# Quels sont les trois cas qui font gagner l'être humain ?
cas_gagnant_1 = (choix_humain == 'pierre'  and choix_ordinateur == 'ciseaux')
cas_gagnant_2 = (choix_humain == 'feuille' and choix_ordinateur == 'pierre')
cas_gagnant_3 = (choix_humain == 'ciseaux' and choix_ordinateur == 'feuille')

if choix_humain == choix_ordinateur :
    print('Vainqueur : égalite')
elif cas_gagnant_1 or cas_gagnant_2 or cas_gagnant_3:
    print('Vainqueur : humain')
else :
    print('Vainqueur : ordinateur')
```

Exercice 5 — Le juste prix (⋆ ⋆ ⋆)

Le juste prix consiste à deviner le prix d'un objet (un entier inférieur ou égal à 100 euros) en faisant des propositions de prix, l'autre joueur indiquant si on est au-dessus ou en dessous du prix réel. On commencera par créer un fichier `exercice5.py`.

1. En vous inspirant de l'exercice précédent, programmez le jeu du juste prix où c'est l'humain qui doit deviner le prix. Il a droit à 7 propositions. Indication : `int('42') == 42`.
   On définit une fonction auxiliaire `juste_prix_essais(n)` ou `n` correspond aux nombres de tentatives restantes. Si $n = 0$, il n'y a plus de tentatives. Au départ $n = 7$. Cette fonction auxiliaire renvoie un booléen qui dit si on a trouvé (`True`) ou pas (`False`).

```python
def essai(prix):
    pronostic = int(input('Quel est votre pronostic ?'))
    if pronostic == prix:
        print('Trop bien !')
        return True
    elif pronostic > prix:
        print('Trop haut !')
        return False
    else:
        print('Trop bas !')
        return False


def juste_prix_essais(n,prix):
    if n == 0:
        return False
    else:
        tentative = essai(prix)
        if tentative:
            return True
        else:
            return juste_prix_essais(n-1,prix)

# On peut écrire juste_prix_essais de manière plus dense.
# À vous de choisir la méthode qui vous semble la plus simple à comprendre.
def juste_prix_essais(n,prix):
    if n == 0:
        return False
    else:
        return essai(prix) or juste_prix_essais(n-1,prix)



def juste_prix():
    prix = randint(0,100)
    trouvé = juste_prix_essais(7,prix)
    if trouvé:
        print('Gagné !')
    else:
        print('Essais épuisés, perdu, la valeur était', prix,trouvé)
```

2. Quelle stratégie adoptez-vous pour gagner à coup sûr ?
   La stratégie à adopter est une dichotomie, on connaît les valeurs les plus grandes et petites possibles, on teste au milieu. *7 essais suffisent car la longueur de l'intervalle est divisée par 2 à chaque étape. En partant d'un intervalle de longueur 100, on obtient au plus un intervalle réduit à un élément en 7 étapes.*

3. Programmez le jeu où c'est l'ordinateur qui fait des propositions et l'humain qui après avoir choisi un nombre entre 0 et 100 répond « + » ou « − » selon la valeur proposée par l'ordinateur. Si l'ordinateur a trouvé la solution, vous répondrez « = » L'ordinateur a droit à 7 propositions et devra gagner à coup sûr ou afficher un message spécial si l'humain a triché dans ses réponses.

```python
def essai_ordinateur(valeur_min, valeur_max):
    if valeur_min > valeur_max:
        return False
    essai = (valeur_min + valeur_max) // 2
    print('Je devine', essai)
    estimation = input()
    if estimation == '=':
        return True
    elif estimation == '+':
        return essai_ordinateur(essai + 1, valeur_max)
    else: # estimation == '-'
        return essai_ordinateur(valeur_min, essai - 1)


def juste_prix2():
    trouvé = essai_ordinateur(0,100)
    if not trouvé:
        print('Tricheur !')
```