

# Premiers pas avec IDLE, premier jeu

Étienne Lozes – Université Nice Sophia Antipolis

Formation ISN Python 2018

## Objectifs:

- Apprendre à manipuler le toplevel de Idle
- Utiliser Python comme une calculatrice évoluée
- Savoir écrire et exécuter un programme avec Idle
- Jeu pierre-feuille-ciseaux

## Avant de commencer

Sur votre ordinateur, créez un répertoire Python dans lequel vous rangerez tous vos fichiers de la formation.

## Lisibilité

Il est de bonne pratique de choisir une police de taille fixe (Courier ou Monaco) et suffisante (16 par exemple) pour ne pas fatiguer les yeux, d'avoir un fond d'écran non blanc (jaune pâle par exemple), et de laisser un espace de part et d'autre d'un opérateur : il est plus facile de lire  $2 - x + 3 // y$  que  $2-x+3//y$  si l'on est à 1 mètre d'un écran !

## Exercice 1 (Premiers pas dans le toplevel, ☆)

1. Définissez deux variables :  $p$  ayant pour valeur 5 et  $q$  ayant pour valeur  $3p$ .
2. Calculez  $(p + \lfloor \frac{q}{2} \rfloor)^2$ , où  $\lfloor r \rfloor$  désigne le plus grand entier  $n \leq r$  (la "partie entière" de  $r$ ).
3. Demander en python «  $q$  est-il un multiple de  $p$  ? ». Indication : utilisez le reste de la division.
4. Traduisez en Python la phrase suivante :  
« si  $q$  est un multiple de  $p$ , afficher OUI sinon afficher NON »
5. Sans consulter vos notes de cours, écrivez au toplevel les lignes de code permettant d'échanger les valeurs de  $p$  et  $q$ , en utilisant une variable temporaire `tmp`. Vérifiez ensuite que les valeurs de  $p$  et  $q$  ont bien été permutées.
6. Calculez la moyenne des entiers de 1 à 10. Le résultat est un nombre flottant (à virgule).
7. Calculez la moyenne des entiers de 1 à 11. Ne retaper pas la formule complète, mais modifiez la précédente en tapant **Alt-p** (**Ctl-p** sur Mac) qui va chercher les lignes précédemment entrées (l'historique). On utilise **Alt-n** (**Ctl-n** sur Mac) en sens inverse.
8. Comment feriez-vous pour savoir si la fraction  $51/85$  est irréductible ? En d'autres termes, peut-on la simplifier ? Par combien ?
9. Demandez en une ligne si  $5^4$  est plus grand que  $4^5$ .
10. Tapez au toplevel : `help(max)` pour demander une petite doc de la fonction `max`

Importez la fonction `randint` en écrivant `from random import randint`. Évaluez plusieurs fois `randint(1)`, puis écrivez `x=randint(1)` et évaluez plusieurs fois `x`. Que constatez-vous ?

Définissez une fonction `cri(n)` qui affiche `n` suivi de trois points d'exclamation et qui renvoie `n`.

Évaluez `1+cri(2)`. Vous devriez voir s'afficher

```
1 2!!!
2 3
```

□

### Exercice 2 (Convertir l'heure, \*\*)

Écrivez une fonction `hconv(n)` prenant un entier  $n > 0$  représentant un nombre de secondes. L'effet de cette fonction sans résultat est l'affichage d'une ligne exprimant la conversion de  $n$  secondes en heures-minutes-secondes. Exemple :

```
1 >>> hconv(4567)
2 4567 --> 1 heure(s) 16 minute(s) 7 seconde(s)
```

Faites ensuite l'accord du pluriel correctement ! Exemple :

```
1 >>> hconv(4560)
2 4560 --> 1 heure 16 minutes
3 >>> hconv(3607)
4 4567 --> 1 heure 7 secondes
```

□

## Du toplevel aux vrais programmes

Un vrai programme est une suite d'instructions lue et exécutée d'un seul bloc et enregistrée dans un fichier. Vous allez écrire votre premier programme dans l'éditeur de Idle. Dans le menu, choisissez **File/New File**, puis tapez votre programme dans la nouvelle fenêtre, et quand vous êtes prêts, exécutez votre programme en choisissant **Run/Run Module**.

Lorsque l'on vous demandera de sauver le contenu de votre éditeur, vous opterez pour un fichier de nom `mourre.py`, dans votre dossier nommé Python.

### Exercice 3 (Pierre-Feuille-Ciseaux, \*\*)

Le jeu pierre-feuille-ciseaux (aussi appelé parfois mourre en Italie ou 手勢令 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. Écrivez un programme qui affiche la phrase « quel est ton choix, humain? » puis qui demande à l'utilisateur de saisir un texte, et enfin affiche la phrase « tu as choisi XXXX, humain », où XXXX est remplacé par le texte saisi par l'utilisateur juste avant. Par exemple :

```
1     quel est ton choix, humain? pierre
2     tu as choisi pierre, humain
3
```

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.

2. 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 choisis pour toi ». Par exemple, on aura :

```
1   quel est ton choix, humain? pierre
2   ok
3
```

*ou encore*

```
1   quel est ton choix, humain? puits
2   choix incorrect, je choisis pour toi ...
3
```

3. 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). Complétez votre programme pour que le message d'erreur précédant indique le choix qu'a fait l'ordinateur. On aura désormais

```
1   quel est ton choix, humain? puits
2   choix incorrect, je choisis pour toi : feuille
3
```

4. 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 :

```
1   quel est ton choix, humain? ciseaux
2   ok
3   humain : ciseaux
4   ordi : feuille
5   gagnant : humain
6
```

□