

Informatique – Biologie L2 – TP1 : Programmation Python

Jean-Baptiste LAMY (jiba@tuxfamily.org)

5 février 2007

1 Exercice de base

Lancer l'environnement de programmation IDLE. Utiliser l'interpréteur Python en mode console pour effectuer des calculs et manipuler les différents types de données.

Réponse :

Calculs : utiliser l'interpréteur Python comme une calculatrice. Montrer les avantages de l'interpréteur Python en mode console : chaque instruction est immédiatement évaluée, permettant un rapide débogage.

Assignements de valeurs simples dans des variables.

Types : nombres, textes, booléens. A chaque fois, montrer comment utiliser la commande 'print' pour connaître le contenu d'une variable.

Même chose, mais avec des listes : montrer que chaque valeur stockée dans une liste reçoit un index, fonction de sa position. Insister sur le fait que l'index du premier élément est 0, celui du n-ième n-1. Montrer la syntaxe employée pour accéder à une valeur grâce à son index (lecture, écriture).

Même chose, mais avec des dictionnaires : montrer que chaque valeur stockée est associée à une clef. Montrer la syntaxe employée pour accéder à une valeur grâce à sa clef.

2 Calcul de l'indice de masse corporelle

L'indice de masse corporelle d'une personne peut être calculé avec la formule suivante :

$$\text{imc} = \text{poids} / (\text{taille}^2)$$

Écrire un programme qui demande à l'utilisateur les paramètres nécessaires, calcule l'indice de masse corporelle et affiche le résultat. Quels sont les paramètres ? Quels sont leur type ? Comment calculer taille^2 ?

NB : vous pouvez vous aider du programme réalisé en TD/cours.

3 Calcul de la clairance rénale

On souhaite réaliser un programme estimant la clairance à la créatinine à partir de la créatinine plasmatique (ou créatinémie). Cette estimation peut être réalisée avec la formule de Cockcroft (voir ci-dessous). Nous allons réaliser un programme en ligne de commande ; des logiciels de ce type sont couramment utilisés par les biologistes dans les hôpitaux.

NB : vous pouvez vous aider du programme réalisé en TD/cours.

Réponse :

```
poids = float(raw_input("Quel est votre poids (kg) ? "))
taille = float(raw_input("Quel est votre taille (m) ? "))

imc = poids / (taille * taille)

print "indice de masse corporelle :", imc
```

3.1 Rappel

Pour rappel voici la formule :

Pour l'homme : Clairance (en ml/min) = $1,24 \times (140 - \text{age}) \times \text{poids (en kg)} / \text{créatinine (en } \mu\text{mol/l)}$

Pour la femme : Clairance (en ml/min) = $1,04 \times (140 - \text{age}) \times \text{poids (en kg)} / \text{créatinine (en } \mu\text{mol/l)}$

Limites de validité :

– Âge : 18 à 110 ans

– Poids : 35 à 120 kg

– Créatinine : 50 à 800 $\mu\text{mol/L}$ (si le patient a une créatinine basse, cela peut conduire à une surestimation de la clairance)

Référence : Cockcroft D.W., M.H. Prediction of creatinine clearance from serum creatinine. Nephro 1976 16 :31-41.

3.2 Quels sont les variables à utiliser ?

Pour chaque variable, indiquer son type.

Réponse : Les variables sont :

- sexe : chaîne de caractères, "H" pour un homme et "F" pour une femme,
- âge : entier,
- poids : flottant,
- créatinine plasmatique : flottant.

3.3 Écrire un programme Python pour estimer la clairance

Réponse : Commencer d'emblée un nouveau programme (menu File -> New window dans IDLE) sans utiliser le "mode immédiat". Puis exécuter le programme avec le menu Run -> Run module.

```
sexe = raw_input("Sexe du patient (H ou F) ? ")
age = int (raw_input("Âge du patient ? "))
poids = float(raw_input("Poids du patient (en Kg) ? "))
creatinine_plasmatique = float(raw_input("Créatinine plasmatique (en µmol/l) ? "))

if sexe == "H" :
    clairance = (140 - age) * poids * 1.24 / creatinine_plasmatique
else :
    clairance = (140 - age) * poids * 1.04 / creatinine_plasmatique

print "Clairance de la créatinine :", clairance, "ml/mn"
```

3.4 Utiliser votre programme pour calculer la clairance d'un homme de 54 ans, pesant 60 kg et ayant une créatininémie de 90µM/l

Réponse : 71.1 ml/min

4 Un programme pour identifier des orchidées

Il existe un grand nombre d'espèces d'orchidées. Celles-ci peuvent être identifiées à partir de critères morphologiques ou environnementaux (type de milieu,...). Notre objectif est d'écrire un programme permettant d'identifier une orchidée en demandant les différents critères à l'utilisateur.

Dans le TP, nous nous limiterons aux familles, genres et espèces suivants :

- Cyripédiacées : 2 étamines fertiles
 - Sabot de Vénus (une seule espèce)
- Orchidacées : 1 étamine fertile
 - Epipactis : rhizome
 - Helleborine des marais : milieu humide (marais)
 - Épipactis à larges feuilles : milieu sec, fleurs vertes
 - Épipactis rouge sombre : milieu sec, fleurs rouges
 - Dactylorhiza : tubercule digité
 - Dactylorhiza de mai : tige creuse, fleurs pourpres
 - Orchis tacheté : fleurs blanches ou roses

4.1 Écrire un programme pour identifier la famille de l'orchidée

Quelles sont les familles présentes dans la classification ci-dessus ?

Le programme doit poser une / des question(s), et ensuite indiquer à l'écran à quelle famille appartient l'orchidée.

Réponse : Les familles sont Cyripédiacées et Orchidacées.

Dans le programme, faire attention lorsque l'on demande le nombre d'étamine : penser à le convertir en entier !

```
nb_etamines = int(raw_input("Nombre d'étamines fertiles ? "))

if nb_etamines == 2 :
    print "Cyripédiacées"

elif nb_etamines == 1 :
    print "Orchidacées"
```

4.2 Écrire un programme pour identifier une orchidée

Le programme doit poser les questions correspondant aux différents critères. Essayer de poser le minimum de questions à l'utilisateur (par exemple, si l'orchidée a 2 étamines fertiles, ce n'est pas la peine de demander la couleur de la fleur!). Lorsque l'orchidée est identifiée, écrire à l'écran le nom de l'espèce.

Réponse : Attention aux ET et aux OU.

```
nb_etamines = int(raw_input("Nombre d'étamines fertiles? "))
if nb_etamines == 2 :
    print "Sabot de Vénus"

elif nb_etamines == 1 :
    racine = raw_input("Type de racine (tubercule ou rhizome)? ")

    if racine == "rhizome" :
        milieu = raw_input("Type de milieu (humide ou sec)? ")
        if milieu == "humide" :
            print "Helleborine des marais"

        else :
            couleur = raw_input("Couleur des fleurs? ")

            if couleur == "vert" :
                print "Épipactis à larges feuilles"

            elif couleur == "rouge" :
                print "Épipactis rouge sombre"
    elif racine == "tubercule" :
        tige_creuse = raw_input("Tige creuse (oui ou non)? ")
        couleur = raw_input("Couleur des fleurs? ")
        if (tige_creuse == "oui") and (couleur == "pourpre") :
            print "Dactylorhiza de mai"

    elif (couleur == "blanc") or (couleur == "rose") :
        print "Orchis tacheté"
```

4.3 Identifier une orchidée

Utiliser votre programme pour identifier les orchidées suivantes :

1. Une orchidée avec 2 étamines fertiles et une fleur jaune
2. Une orchidée avec 1 étamine fertile, des tubercules et une fleur rose
3. Une orchidée avec 1 étamine fertile, un rhizome et une fleur violette, poussant en milieu sec



4.4 Écrire un programme pour identifier une orchidée (2)

Modifier le programme précédent pour que, lorsque l'orchidée n'a pas pu être identifiée, il écrive à l'écran "vous avez trouvé une nouvelle espèce d'orchidée!".

Réponse : Pour cela, il faut créer une variable pour stocker si oui ou non une espèce a été trouvée.

```
espece = ""
nb_etamines = int(raw_input("Nombre d'étamines fertiles? "))
if nb_etamines == 2 :
    espece = "Sabot de Vénus"
```

```

elif nb_etamines == 1 :
    racine = raw_input("Type de racine (tubercule ou rhizome)? ")

    if racine == "rhizome" :
        milieu = raw_input("Type de milieu (humide ou sec)? ")
        if milieu == "humide" :
            espece = "Helleborine des marais"

        else :
            couleur = raw_input("Couleur des fleurs? ")

            if couleur == "vert" :
                espece = "Épipactis à larges feuilles"

            elif couleur == "rouge" :
                espece = "Épipactis rouge sombre"
    elif racine == "tubercule" :
        tige_creuse = raw_input("Tige creuse (oui ou non)? ")
        couleur = raw_input("Couleur des fleurs? ")
        if (tige_creuse == "oui") and (couleur == "pourpre") :
            espece = "Dactylorhiza de mai"

        elif (couleur == "blanc") or (couleur == "rose") :
            espece = "Orchis tacheté"

if espece == "" :
    print "Vous avez trouvé une nouvelle espèce d'orchidée!"

else :
    print "C'est un(e)", espece

```

4.5 Écrire un programme pour identifier une orchidée (3)

Modifier le programme précédent pour qu'il permette d'identifier plusieurs orchidées sans avoir à relancer le programme. Après la première identification, le programme doit demander si l'on souhaite identifier une nouvelle orchidée, et si oui, recommencer le programme.

Réponse : Il faut faire une boucle while; attention à mettre la (re)définition de la variable espèce dans la boucle!

```

while 1 :
    espece = ""

    nb_etamines = int(raw_input("Nombre d'étamines fertiles? "))
    if nb_etamines == 2 :
        espece = "Sabot de Vénus"

    elif nb_etamines == 1 :
        racine = raw_input("Type de racine (tubercule ou rhizome)? ")
        if racine == "rhizome" :
            milieu = raw_input("Type de milieu (humide ou sec)? ")
            if milieu == "humide" :
                espece = "Helleborine des marais"

            else :
                couleur = raw_input("Couleur des fleurs? ")
                if couleur == "vert" :
                    espece = "Épipactis à larges feuilles"

                elif couleur == "rouge" :
                    espece = "Épipactis rouge sombre"

        elif racine == "tubercule" :
            tige_creuse = raw_input("Tige creuse (oui ou non)? ")
            couleur = raw_input("Couleur des fleurs? ")
            if (tige_creuse == "oui") and (couleur == "pourpre") :
                espece = "Dactylorhiza de mai"

            elif (couleur == "blanc") or (couleur == "rose") :
                espece = "Orchis tacheté"

```

```

if espece == "" :
    print "Vous avez trouvé une nouvelle espèce d'orchidée !"
else :
    print "C'est un(e)", espece

encore = raw_input("Identifier une nouvelle orchidée (oui ou non)? ")
if encore == "non" : break

```

4.6 Écrire un programme pour identifier une orchidée et indiquer le type d'exposition

Modifier le programme précédent pour qu'il indique, après identification de l'orchidée, le type d'exposition conseillé pour faire pousser cet orchidée, sachant que l'Helleborine des marais, le Dactylorhiza de mai et l'Orchis tacheté sont des espèces de pleine lumière, alors que le Sabot de vénus et les Épipactis préfèrent l'ombre.

Faire en sorte que l'exposition soit indiquée même si l'orchidée n'a pas pu être entièrement identifiée, si les données le permettent (c'est le cas dans l'orchidée n°3 de la question 3.3).

Réponse :

```

while 1 :
    espece = ""
    exposition = ""

    nb_etamines = int(raw_input("Nombre d'étamines fertiles? "))

    if nb_etamines == 2 :
        exposition = "ombre"
        espece = "Sabot de Vénus"

    elif nb_etamines == 1 :
        racine = raw_input("Type de racine (tubercule ou rhizome)? ")

        if racine == "rhizome" :
            milieu = raw_input("Type de milieu (humide ou sec)? ")

            if milieu == "humide" :
                espece = "Helleborine des marais"
                exposition = "soleil"

            else :
                exposition = "ombre"

            couleur = raw_input("Couleur des fleurs? ")

            if couleur == "vert" :
                espece = "Épipactis à larges feuilles"

            elif couleur == "rouge" :
                espece = "Épipactis rouge sombre"

        elif racine == "tubercule" :
            exposition = "soleil"

            tige_creuse = raw_input("Tige creuse (oui ou non)? ")
            couleur = raw_input("Couleur des fleurs? ")

            if (tige_creuse == "oui") and (couleur == "pourpre") :
                espece = "Dactylorhiza de mai"

            elif (couleur == "blanc") or (couleur == "rose") :
                espece = "Orchis tacheté"

    print "L'exposition conseillée est :", exposition
    if espece == "" :
        print "Vous avez trouvé une nouvelle espèce d'orchidée !"

    else :
        print "C'est un(e)", espece

    encore = raw_input("Identifier une nouvelle orchidée (oui ou non)? ")

```

```
if encore == "non" : break
```