

# LS4 - Travaux pratiques 1

20 janvier 2014

## 1 Prendre le langage en main

Vous pouvez consulter le tutoriel <sup>1</sup>.

Répondre aux questions suivantes en évaluant les instructions à la main puis en s'aidant d'un interpréteur interactif.

### 1.1 Variables et affectations

#### Question 1

Déterminer les valeurs des variables à l'issue des lignes de commandes suivantes.

```
ai = 1
i = 1
ai = 3
i = 3
ai = 6
```

```
ai :
i :
```

```
a, b = 2, 4
c = a
b = a
c = b - a
```

```
a :
b :
c :
```

#### Question 2

Dans un fichier python, écrire une suite d'instructions qui permet d'échanger le contenu de deux variables entières x et y :

1. à l'aide d'une variable intermédiaire tmp ;
2. sans variable intermédiaire.

---

1. <http://docs.python.org/3/tutorial/introduction.html>

### Question 3

Interpréter la suite d'instructions suivantes. Trouver les erreurs et leur associer un message. À l'adresse (suivante<sup>2</sup>) vous trouverez les erreurs fréquentes.

```
a = 4
b = 2
2 = a
b = 3
```

- ☐ Un entier n'est pas une variable
- ☐ Une variable qui n'a pas été définie n'a pas de valeur
- ☐ Confusion minuscule et majuscule dans l'identificateur
- ☐ L'affectation est une opération dirigée (a=b n'est pas la même chose que b=a)

```
a = 2
b = A
```

- ☐ Un entier n'est pas une variable
- ☐ Une variable qui n'a pas été définie n'a pas de valeur
- ☐ Confusion minuscule et majuscule dans l'identificateur
- ☐ L'affectation est une opération dirigée (a=b n'est pas la même chose que b=a)

## 1.2 Int et Float

Références à consulter : [functions](#)<sup>3</sup>, [math](#)<sup>4</sup>, [floatingpoint](#)<sup>5</sup>.

### Question 1

Les expressions suivantes s'évaluent-elles en des Int ou en des Float ?.

```
3/4
```

- ☐ Int
- ☐ Float

```
4/2
```

- ☐ Int
- ☐ Float

```
3//4
```

- ☐ Int
- ☐ Float

```
4//2
```

- ☐ Int
- ☐ Float

---

2. [http://liris.cnrs.fr/~crigotti/python/source90\\_ERREURS\\_COMMUNES.html](http://liris.cnrs.fr/~crigotti/python/source90_ERREURS_COMMUNES.html)

3. <http://docs.python.org/3.3/library/functions.html>

4. <http://docs.python.org/3.3/library/math.html>

5. <http://docs.python.org/3.3/tutorial/floatingpoint.html>

3.14

- ☐ Int  
☐ Float

10.

- ☐ Int  
☐ Float

1e10

- ☐ Int  
☐ Float

3 \*\* 2

- ☐ Int  
☐ Float

3 \*\* (1/2)

- ☐ Int  
☐ Float

Quel est le résultat du calcul :

r = 19875+77569 % 7

- ☐ 4  
☐ 19877

### Question 3

Calculer la formule suivante en utilisant les fonctions du module `math`.

$$\sqrt{3} + \frac{56}{9.0} \cdot \left| -\frac{1}{4} \right|$$

### Question 4

Quels sont les résultats des calculs suivants :

.1+.1+.1 == .3

- ☐ True  
☐ False

0.10000000000000001 == 0.1

- ☐ True  
☐ False

round(.1,1) + round(.1,1) + round(.1,1) == round(.3,1)

- ☐ True  
☐ False

round(.1 + .1 + .1, 1) == round(.3, 1)

- ☐ True  
☐ False

## 1.3 Fonctions

### Question 1

Expliquer pourquoi les programmes suivants ne fonctionnent pas.

```
rad_deg def(angle) :  
    return (angle * 180 / 3.1)
```

- ☐ Problème d'indentation
- ☐ Mauvaise utilisation du mot clef def
- ☐ Utilisation d'une variable non définie
- ☐ Confusion test (==) et affectation (=)

```
def vol(r,l) :  
    return (pi * r**2 * l)
```

- ☐ Problème d'indentation
- ☐ Mauvaise utilisation du mot clef def
- ☐ Utilisation d'une variable non définie
- ☐ Confusion test (==) et affectation (=)

```
def vol(r,l) :  
    aire = 3.14 * r**2  
    volume = aire * l  
    return volume
```

- ☐ Problème d'indentation
- ☐ Mauvaise utilisation du mot clef def
- ☐ Utilisation d'une variable non définie
- ☐ Confusion test (==) et affectation (=)

```
def vol(r,l,volume) :  
    return (volume = pi * r**2 * l)
```

- ☐ Problème d'indentation
- ☐ Mauvaise utilisation du mot clef def
- ☐ Utilisation d'une variable non définie
- ☐ Confusion test (==) et affectation (=)

### Question 2

Évaluer à la main puis à l'aide de l'interpréteur.

```
def f(y):  
    x = 0  
    while (y != 0):  
        x = x + 3  
        y = y - 1  
    return x
```

```
f(0) :  
f(3) :  
f(5) :  
f(8) :
```

### Question 3

Évaluer à la main les variables locales lors de l'évaluation `fi(5)`. Donner les résultats des évaluations.

```
def f1(x):  
    y = 2 * x  
    x = x + 3  
    x = y - x  
    return (x,y)  
  
def f2(x):  
    y = 2 * x  
    x = y - x  
    x = x + 3  
    return (x,y)  
  
def f3(x):  
    x = x + 3  
    y = 2 * x  
    x = y - x  
    return (x,y)
```

```
f1(5)  :  
f2(5)  :  
f3(5)  :
```

### Question 4

Evaluer chacune des variables globales suivantes :

```
Flag = True  
  
def f(x,y):  
    global Flag  
    x = x - 2  
    Flag = (x > y)  
    return (x, y, Flag)  
  
(a, b, c) = f(7, 2)  
(r, s, t) = f(2, 2)
```

```
Flag    :  
(a, b, c) :  
(r, s, t) :
```

```

Flag = True

def f(x,y):
    x = x - 2
    Flag = (x > y)
    return (x, y, Flag)

(a, b, c) = f(7, 2)
(r, s, t) = f(2, 2)

```

```

Flag      :
(a, b, c)  :
(r, s, t)  :

```

```

x = 5

def f(y):
    global x
    x = 2 * y
    y = x - 1
    return (x, y)

(a, b) = f(7)
(r, s) = f(0)

```

```

x      :
(a, b)  :
(r, s)  :

```

```

x = 5

def f(y):
    x = 2 * y
    y = x - 1
    return (x, y)

(a, b) = f(7)
(r, s) = f(0)

```

```

x      :
(a, b)  :
(r, s)  :

```

```

x = 5

def f(x):
    x = 2 * x
    y = x - 1
    return (x, y)

(a, b) = f(7)
(r, s) = f(0)

```

x	:
(a, b)	:
(r, s)	:

### Question 5

Pour chacun des programmes suivants, déterminer les valeurs de `mult_3(4)` puis de la variable globale `N`. Si `N` n'existe pas, écrire **erreur**. Lesquels de ces programmes permettent de retourner la multiplication par 3 ? Dire pourquoi certains programmes ne répondent pas la bonne valeur :

<pre>N = 3 def mult_3(x) :     N = 4     return (N * x)</pre>
---

- ☐ Redéfinition locale d'une variable globale
- ☐ Indentation
- ☐ Un argument est une variable locale
- ☐ La fonction calcule la multiplication par 3

<code>mult_3 (4)</code>	:
<code>N</code>	:

<pre>N = 3 def mult_3(N) :     return (N * N)</pre>
---

- ☐ Redéfinition locale d'une variable globale
- ☐ Indentation
- ☐ Un argument est une variable locale
- ☐ La fonction calcule la multiplication par 3

<code>mult_3 (4)</code>	:
<code>N</code>	:

<pre>def mult_3(N) :     N = 3     return (N * N)</pre>
---

- ☐ Redéfinition locale d'une variable globale
- ☐ Indentation
- ☐ Un argument est une variable locale
- ☐ La fonction calcule la multiplication par 3

<code>mult_3 (4)</code>	:
<code>N</code>	:

<pre>def mult_3(x) :     N = 3     return (N * x)</pre>
---

- ☐ Redéfinition locale d'une variable globale
- ☐ Indentation
- ☐ Un argument est une variable locale
- ☐ La fonction calcule la multiplication par 3

<code>mult_3 (4)</code>	:
<code>N</code>	:

## 1.4 Conditionnelles

Pour chacune des fonctions `f` suivantes, déterminer la valeur de `f(4)`.

```
def f(x) :  
    if x % 2 == 0 :  
        a = 0  
    elif x % 4 == 0 :  
        a = 1  
    else :  
        a = 3  
    return a
```

`f(4)` :

```
def f(x) :  
    if x % 2 == 0 :  
        a = 0  
    else:  
        a = 1  
    return a
```

`f(4)` :

```
def f(x) :  
    if x % 2 == 0 :  
        a = 0  
    if x % 4 == 0 :  
        a = 1  
    return a
```

`f(4)` :

```
def f(x) :  
    if x % 2 == 0 :  
        return 0  
    if x % 4 == 0 :  
        return 1
```

`f(4)` :

### Question 2

Écrire une fonction `ordre(x, y, z)` qui renvoie `True` si `x` plus petit que `y` et `y` plus petit que `z` et `False` sinon.

### Question 3

Écrire une fonction `max(x,y)` qui calcule le max de `x` et de `y`.



## 1.5 Boucles

Dire ce qui ne convient pas dans les programmes suivants.

```
x=3
while x=3:
    if x <= 4:
        x+2
    else:
        return x
```

- ☐ Comparaison == ou != avec flottants (boucle qui ne termine pas)
- ☐ Utilisation de = à la place de == (invalid syntax)
- ☐ Indentation

```
x = 0
while x != 25 / 3:
    x = x + 1
```

- ☐ Comparaison == ou != avec flottants (boucle qui ne termine pas)
- ☐ Utilisation de = à la place de == (invalid syntax)
- ☐ Indentation

```
def carre(x):
    c=0
    y=x
    while c<x :
        y=y+x
    return y
```

- ☐ Comparaison == ou != avec flottants (boucle qui ne termine pas)
- ☐ Utilisation de = à la place de == (invalid syntax)
- ☐ Indentation

## 1.6 Scripts

Qu'est-ce qu'un exécutable ?

- ☐ un programme que l'on peut exécuter
- ☐ un fichier qui a les droits en exécution
- ☐ un répertoire
- ☐ un fichier qui a les droits en écriture

Quelle commande permet de donner les droits à l'utilisateur en lecture et exécution au fichier script.py ?

- ☐ chmod u+rx script.py
- ☐ chmod u+w script.py
- ☐ chown user script.py
- ☐ chmod 755 script.py

Que signifie la ligne suivante :

```
#!/usr/bin/env python3
```

- ☐ l'adresse du programme à utiliser pour exécuter le script
- ☐ le nom du langage du cours
- ☐ l'adresse du script exécuté

Quelle commande shell pour demander où est python ?

- ☐ what python3
- ☐ which python3
- ☐ who python3

Comment exécuter le script `mon_script.py` ?

- ☐ `./mon_script`
- ☐ `./mon_script.py`
- ☐ `mkdir ~/my_bin puis cp mon_script.py ~/my_bin` éditer le fichier `~/bashrc` et modifier la variable `PATH`.

## 2 Affichage d'horloge

### 2.1 Fonctions

#### Question 1

Écrire une fonction `sec_to_hms` qui convertit un temps exprimé en seconde en un temps exprimé en heure :minute :seconde.

Par exemple, `sec_to_hms(65)` s'évalue en `(0,1,5)`, `sec_to_hms(3601)` s'évalue en `(1,0,1)`.

#### Question 2

Écrire la fonction opposée `hms_to_sec`. Par exemple, `sec_to_hms(1, 0, 5)` s'évalue en `65`, `sec_to_hms(1, 0, 1)` s'évalue en `3601`.

En utilisant les fonctions `strftime` et `localtime` du module `time`, écrire une fonction `now` telle que `now()` retourne une chaîne de caractères représentant la date et l'heure de l'ordinateur sous le format `%Y-%m-%d %H:%M:%S`.

Par exemple, si il est minuit, cinq minutes et trente secondes, le premier janvier 2014, `now()` s'évalue en `2014-01-01 00:05:30`.

### 2.2 Scripts et modules

Télécharger le script suivant et le compléter.

`my_time.py`<sup>6</sup>

#### Question 1

À quoi sert la variable `__name__`, quelles sont ses valeurs possibles ?

- ☐ le nom de l'utilisateur
- ☐ le nom du script
- ☐ la chaîne de caractère `"__name__"`

#### Question 2

Compléter le script pour qu'exécuté dans un terminal, il affiche sur la sortie standard l'heure actuelle en hms en la mettant à jour toutes les 5s.

### 2.3 Interface graphique

Télécharger une interface graphique, la compléter.

---

6. <http://didascalie.informatique.univ-paris-diderot.fr:80>

### Question 1

Faire une horloge digitale en complétant l'interface graphique suivante.  
horloge\_digital.py<sup>7</sup>

### Question 2

Faire une horloge à cadran en complétant l'interface graphique suivante.  
horloge\_cadran.py<sup>8</sup>

## 2.4 Bonus

Faire un script et une interface graphique implémentant un chronomètre.

---

7. <http://didascalie.informatique.univ-paris-diderot.fr:80>

8. <http://didascalie.informatique.univ-paris-diderot.fr:80>