

12 TP N° 12

Dans les TP précédents, nous avons étudié la boucle Tant que (`while`) et la structure conditionnelle c'est à dire *si ... , alors ... , sinon ...* (`if, else`).

Ces deux types d'instruction font intervenir une condition, par exemples :

- Tant que $p < 23\,000$, la condition est : $p < 23\,000$
- Si $\text{age} < 50$, la condition est : $\text{age} < 50$.

Ces conditions sont écrites avec des signes comme le signe $<$ ou le signe $=$. Le tableau ci-dessous donne la correspondance entre chaque signe mathématique et sa traduction en Python.

| | | | | | | |
|-------------------------------|----|---|----|---|----|----|
| Signe mathématique | = | < | ≤ | > | ≥ | ≠ |
| Signe correspondant en Python | == | < | <= | > | >= | != |

Tab. 12.1

Exercice 12.1 (Vrai, faux, variable booléenne)

1. Aller dans la console, et saisir `1==1`, puis `1==2`, on obtient alors les affichages ci-dessous.

```
>>> 1==1
True
>>> 1==2
False
```

Cela est normal puisque l'égalité $1 = 1$ est vraie, et l'égalité $1 = 2$ est fausse.

2. Toujours dans la console, essayer successivement : `5>=8` `5>=5` `5>=3` `4!=4` `4!=6`
3. En fait, une variable peut prendre la valeur Vrai ou la valeur Faux, et il s'agit alors d'un nouveau type de variable.

a) Pour que la variable a prenne la valeur Vrai ($a \leftarrow \text{Vrai}$), saisir à la console : `>>> a=True`

b) Pour connaître le type de la variable a , saisir à la console : `>>> type(a)`

On voit alors :

```
>>> a=True
>>> type(a)
<class 'bool'>
```

L'affichage `<class 'bool'>` signifie que la variable a est une valeur booléenne.

Définition 12.1 (type de variable bool en Python)

Une variable booléenne est une variable qui a la valeur Vrai ou Faux.

En Python,

- la valeur Vrai est `True`, et la valeur Faux est `False` ;
- une variable booléenne est du type `bool`.

Exercice 12.2

Dans la fonction `dv` et dans l’algorithme ci-dessous, les variables a et b sont des entiers, et la variable `rep` est une variable booléenne.

En Python l’écriture `a%b` est le reste de la division euclidienne de a par b .

| Fonction Python | Algorithme |
|--|--|
| <pre>def dv(a,b): if(a%b==0): rep=True else: rep=False return(rep)</pre> | <p>Si le reste de la division euclidienne de a par b est égal à zéro alors $rep \leftarrow$ Vrai sinon $rep \leftarrow$ Faux Fin du Si</p> |

1. Saisir à la console `dv(42,7)` et `dv(18,7)` et expliquer les réponses obtenues avec le cours d’arithmétique.

.....

2. Pour deux nombres entiers a et b , si la réponse retournée est Vrai, qu’est-ce que ça signifie ?

.....

Exercice 12.3

Dans l’algorithme du tableau 12.2, page suivante, les variables a et d sont des entiers (type `int` en Python).

1. Dans le tableau 12.3, détailler l’exécution de cet algorithme pour $a = 91$ et pour $a = 5$.
 Il y a beaucoup de restes de divisions euclidiennes à calculer, donc on peut aller dans la console et utiliser `%`.

2. Que signifie la valeur finale de la variable d ?

.....

3. Pourquoi est-on certain que la boucle Tant que va s’arrêter ?

.....

4. Saisir la fonction `ppdv` en Python, puis la tester avec $a = 91$ et $a = 5$.

5. Compléter le tableau 12.2.

6. Utiliser la fonction `ppdv` pour compléter le tableau 12.4.

Tab. 12.2

| Algorithme | Fonction Python |
|---|---|
| $d \leftarrow 2$ Tant que le reste de la division euclidienne de a par d est différent de zéro $d \leftarrow d + 1$ | <pre>def ppdv(a): return(d)</pre> |

Tab. 12.3

| $a = 91$ | $a = 5$ |
|---|---|
| $d = \dots\dots\dots$ | $d = \dots\dots\dots$ |
| reste de la division euclidienne de a par $d = \dots\dots\dots$ | reste de la division euclidienne de a par $d = \dots\dots\dots$ |
| $d = \dots\dots\dots$ | $d = \dots\dots\dots$ |
| reste de la division euclidienne de a par $d = \dots\dots\dots$ | reste de la division euclidienne de a par $d = \dots\dots\dots$ |
| $d = \dots\dots\dots$ | $d = \dots\dots\dots$ |
| reste de la division euclidienne de a par $d = \dots\dots\dots$ | reste de la division euclidienne de a par $d = \dots\dots\dots$ |
| $d = \dots\dots\dots$ | $d = \dots\dots\dots$ |
| reste de la division euclidienne de a par $d = \dots\dots\dots$ | reste de la division euclidienne de a par $d = \dots\dots\dots$ |
| $d = \dots\dots\dots$ | $d = \dots\dots\dots$ |
| reste de la division euclidienne de a par $d = \dots\dots\dots$ | reste de la division euclidienne de a par $d = \dots\dots\dots$ |
| $d = \dots\dots\dots$ | $d = \dots\dots\dots$ |
| reste de la division euclidienne de a par $d = \dots\dots\dots$ | reste de la division euclidienne de a par $d = \dots\dots\dots$ |
| $d = \dots\dots\dots$ | $d = \dots\dots\dots$ |

Tab. 12.4

| | | | | | |
|----------------------------------|----|----|----|-----|-----|
| a | 15 | 19 | 77 | 251 | 943 |
| Valeur finale de la variable d | | | | | |

Exercice 12.4

Un nombre premier est un nombre entier qui n'a que deux diviseurs : 1 et lui même. Par exemple 3 est premier parce que ses diviseurs sont 1 et 3, alors que 4 n'est pas premier parce qu'il a 3 diviseurs : 1 ; 2 ; 4. Saisir une fonction Python, nommée **prem**, de variable **a**, qui retourne **True** si **a** est premier et **False** sinon. Dans cette fonction on peut utiliser la fonction **ppdv** de l'exercice 12.4. Recopier cette fonction dans le tableau ci-contre.

| |
|-------|
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |