TD 01 Structures conditionnelles

Lycée Louis Thuillier - Informatique - PCSIB -

1 Tests et opérateurs logiques

1.1 Test et Booléens

Définition. Booléen
Un booléen est une expression qui ne peut prendre que deux valeurs : *True* et *False*.
Ce sont typiquement des test d'égalité ou d'inégalité.

♡ **Commande ♡. Test d'égalité, inégalité, d'infériorité**
- ▷ égalité : ==
- ▷ inégalité : !=
- ▷ ordre : >, <, <= et >=

De tels test sont des booléens : le résultat sera soit vrai, soit faux.

On peut combiner deux booléens A et B par des opérateurs logique
- ▷ *and* : A *and* B est vrai si A et B sont vrais **en même temps**
- ▷ *or* : A *or* B est vrai si **au moins** A ou B est vrai

> *Application 1 : Donner la valeur des Booléens suivants (on donnera évidemment la réponse* ***AVANT*** *de d'exécuter le programme !!)*
>
> ```
> 2+3==6 and 4*0.5>2
>
> 2==5-3 and 3*2!=7
>
> 3*2 != 7 and 2<=6/4 or 2+2==4
> ```

> *Application 2 :* ***Attention ! Important !*** *On va s'en servir toute l'année !!*
> *Donner un Booléen qui permet de vérifier si un nombre n est un multiple d'un nombre m.*

1.2 Instructions conditionnelles

A quoi ça sert les booléens ? Les Booléens permettent de déclencher ou non l'exécution de certaines lignes de commandes : c'est ce qu'on appelle **les instructions conditionnelles**.

Définition. Strucuture conditionnelle
Une structure conditionnelle permet de lister une suite d'instruction qui ne seront lues et exécutées que **si** une condition est vraie.

Exemple 1 : *Les booléens et les structures conditionnels se retrouvent dans la vie de tous les jours.*

"Vas au supermarché acheter une brique de lait. S'ils ont des avocats, prends en trois. Sinon achète des tomates."

On comprend alors dans cet exemple qu'on a utilisé un Booléen, celui qui dit : "le supermarché a des avocats". Cette affirmation est bien soit vraie soit fausse et suivant sa valeur, les courses ne seront pas les mêmes.
On peut l'écrire comme :
- ▷ *SI "le supermarché a des avocats"=TRUE ⇒ acheter 3 avocats*
- ▷ *SINON : acheter des tomates.*

2 Commande if et notion d'indentation

Attention ! *Deux notions importantes vont apparaître dans cette partie et vont se retrouver* ***PARTOUT*** *cette année en informatique : l'écriture d'algorithme sous forme de squelette et la notion d'indentation.*

▶ Structure d'algorithme

Application 3 : *On souhaite écrire un programme qui demande à l'utilisateur son année de naissance et renvoie*
- ▷ *"Vous êtes né une année bissextile : la classe !" si c'est le cas (année bissextile = année multiple de 4)*
- ▷ *"Nul, nul, nul, vous êtes né une année quelconque ..." dans le cas contraire*

On voit que le programme est un peu plus compliqué que ce que nous avons fait depuis lors. On va, avant d'essayer de le coder en Python, dessiner la structure de l'algorithme, c'est-à-dire les différentes étapes qui vont constituer notre programme.

1. *Écrire/Dessiner/... un schéma qui représentera les différentes étapes du programme.*
   Attention ! *Faites le suivant* ***votre*** *façon de penser. Néanmoins, gardez en tête qu'il faudra utiliser des termes d'informatique.*

Le schéma développer représente l'algorithme du programme, *i.e.* les différentes étapes et leurs enchainement qui permettent d'obtenir le résultat souhaité. Pour le moment l'algorithme est "neutre", c'est-à-dire qu'il ne dépend pas du langage informatique qu'on va utiliser.

▶ Indentation

Une des particularité du langage Python (*ce qui, par ailleurs, fait qu'on l'apprécie ou non ...*) est sa structure **visuelle** : en Python le décalage par rapport à la marge, appelée **indentation**, a un sens.
Il y a alors des **structures**, dans le sens géométrique du terme, qu'il s'agit de bien comprendre et apprendre. C'est le cas des instructions, ou structures, conditionnelles.

Définition. Indentation
Une indentation est un décalage verticale par rapport à la marge. Elle s'applique avec la commande *tab* (la double flèche).

```
1 x=2
2     print(x)
3     x=4
4         y=2*x
```

- ▷ *ligne 1 :* indentation nulle
- ▷ *ligne 2 et 3 :* indentée une fois
- ▷ *ligne 4 :* indentée deux fois

Attention ! L'indentation **a un sens :** le code ci-dessus est faux !

▶ Commande *if* et *else*

Le langage de programmation Python a été créé par Guido van Rossum, d'origine hollandaise. Beaucoup de commande et de syntaxe sont empruntées de l'anglais (*heureusement pas du néerlandais !*).
En anglais "si" se dit "if" et "sinon" se dit "else" ....

♡ Commande ♡. Opérateur *if*
Attention ! On fait attention à la **structure** (∼l'indentation des commandes) ! !

```
1  instruction_0
2
3  if condition :
4      instruction_1
5      instruction_2
6
7  else :
8      instruction_3
9
10 instruction_4
```

- ▷ l'instruction_0 sera toujours exécutée avant la structure conditionnelle, l'instruction_4 sera toujours exécutée après la structure conditionnelle

- ▷ *condition* est un booléen qui est soit vrai soit faux. Suivant sa valeur :
  - ▷ *condition* est vraie ⇒ instruction_1 et instruction_2 seront exécutées
  - ▷ *condition* est fausse ⇒ instruction_3 sera exécutée

On peut noter que la commande *else* n'est pas obligatoire (si on n'a rien a faire faire ...).

Attention ! à l'indentation ! !
Pour faire comprendre à Python que les instruction 1 et 2 ne doivent s'exécuter que si la condition est vraie on les **indente** une fois **par rapport** la ligne comportant *if*.

Application 4 :

1. *Réécrire sur votre feuille votre structure d'algorithme en vous forçant à copier la structure (et donc l'indentation) d'une structure conditionnelle if en Python.*
   Attention ! *il ne faut pas mettre les lignes de commandes, juste ce que le programme doit faire.*
2. *A l'aide de la question précédente, écrire un programme qui demande à l'utilisateur son année de naissance et renvoie*
   - ▷ *"Vous êtes né une année bissextile : la classe !" si c'est le cas (année bissextile = année multiple de 4)*
   - ▷ *"Nul, nul, nul, vous êtes né une année quelconque ..." dans le cas contraire*

▶ Structure plus complexe

On peut évidemment (et c'est tout l'intérêt par la suite) imbriquer des structures ensembles. On pourra noter deux façons de faire :

```
1 if condition_1:
2     instruction_1
3 if condition_2:
4     instruction_2
5 else :
6     instruction_3
```

- ▷ si condition_1 est vraie ⇒ instruction_1 s'exécute
- ▷ si condition_2 est vraie ⇒ instruction_2 s'exécute **SAUF** si la première condition précédente a déjà été vérifiée
- ▷ si les deux conditions sont fausses, instruction_3 s'exécute

Pour que l'instruction_2 s'exécute même si la condition_1 est vraie il faut remplacer if condition_2 par elif condition_2.

```
1 if condition_1:
2     if condition_2:
3         instruction_2
4     else :
5         instruction_3
6 else :
7     instruction_4
```

- ▷ si condition_1 est vraie alors :
  - ▷ si condition_2 est vraie ⇒ instruction_2 s'exécute
  - ▷ sinon instruction_3 s'exécute
- ▷ si condition_1 est fausse alors instruction_4 s'exécute

> ***Application 5 :*** *En réalité, une année bissextile est une année multiple de* 4 *mais pas de* 100 *(300 n'est pas bissextile par exemple). On reprend le problème précédent.*
> - ▷ *écrire la structure de l'algorithme en vous forçant à copier la syntaxe de Python (i, else et indentations).*
> - ▷ *Ecrire le programme.*

Exercices d'applications

Ces exercices se traitent soient en fin de TD, soit lorsque vous reprendrez le TD pour réviser.

1. Que va afficher Python après l'exécution des scripts suivants ? (évidemment on ne vérifiera avec l'ordinateur qu'après ! !)
   Checkpoint !

```python
x=1
if x==2 :
    x=3
    print("x est désormais égal à 3")
print(x)
```

```python
x=1
if x==1 :
    x=3
    print("x est désormais égal à 3")
print(x)
```

```python
x=2
if x==2 :
    x=3*2

if x==6:
    x=x/2
    print("pouet")

print(x)
```

```python
x,y=2,4
if x==2 :
    x=1
    if y!=3:
        x=3
if x<2:
    x=x/2
    if y==4:
        x=1

print(x)
```

2. Écrire un programme qui demande à l'utilisateur un nombre x et qui affiche *"Votre nombre est strictement supérieur/inférieur/égal à zéro."* suivant la valeur du nombre donné.
3. Écrire un programme qui demande à l'utilisateur son nom sous la forme d'une chaîne de caractère et affiche *"Bonjour."* si le prénom donné est le votre.
4. Écrire un programme qui demande à l'utilisateur son année de naissance et lui affiche *"Vous êtes né une année paire."* le cas échéant.
5. Écrire un programme qui demande à l'utilisateur une durée en minutes et la converti en heures et minutes.
6. Écrire un programme qui demande à l'utilisateur une durée en secondes et la converti en heures, minutes et secondes.

7. Écrire un programme qui demande à l'utilisateur deux nombres entiers et
   - ▷ renvoie leur somme si leur somme est supérieur strictement à leur produit sauf si le produit est un multiple de 100. A ce moment là, le programme renvoie le produit.
   - ▷ renvoie leur produit si leur produit est supérieur strictement à leur somme et si leur somme est plus grande que 47.
   - ▷ renvoie "Bingo !" si leur somme et produit sont égaux
8. En réalité (*promis c'est bon cette fois*) une année bissextile est une année multiple de 4, sauf les multiples de 100 mais on garde quand même les multiples de 400.
   Exemple : 300 n'est pas bissextile mais 1600 l'est.

   Écrire un programme qui demande à l'utilisateur son année de naissance et renvoie
   - ▷ "Vous êtes né une année bissextile : la classe !" si c'est le cas
   - ▷ "Nul, nul, nul, vous êtes né une année quelconque ..." dans le cas contraire