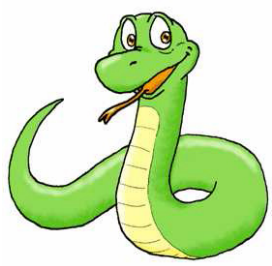
ECT1



TP 05: Les 2 types de structures répétitives

Exercice 1 : Découverte du type liste et de la commande range

Une liste est une série de valeurs, non nécessairement du même type que l'on déclare entre crochets en séparant les différentes valeurs par une virgule.
[] est la liste vide

1- Évaluez dans la console `L=[1,'maison',2<3];L`

2- Évaluez dans la console `L[0]` puis `L[1]` puis `L[2]` puis `L[3]`

Python numérote les indices d'une liste à partir de 0 et non de 1. Ainsi le premier élément de la liste est, pour Python, celui d'indice 0. Le deuxième élément de la liste est, pour Python, celui d'indice 1, etc.

- `L[i]` renvoie la (i + 1)ième coordonnée de L. Ainsi, pour accéder au i ième élément de L, la commande est `L[i-1]`

`range` énumère des éléments mais ne provoque aucun affichage. La liste des éléments énumérés par range s'obtient à l'aide de la commande `list`.

3- Évaluez dans la console `L=list(range(5));L`

4- Qu'énumère les commandes suivantes : `range(1,7)`, `range(1,7,3)`, `range(-2,8,2)`, `range(4,0,-1)`, `range(1,2,0.1)`

Bilan 1: la commande range
range(début , fin , pas)

La commande range ne provoque pas d'affichage et ne crée pas de liste. elle énumère uniquement des entiers de `début` à `fin-1`
Les paramètres *début* et *pas* sont optionnels.

- Dans l'intervalle [0 , fin[si un seul paramètre est renseigné.
`L=list(range(4))` va créer la liste [0 , 1 , 2 , 3] de 4 termes, le premier sera `L[0]=0`, le dernier `L[3]=3`.
- Dans l'intervalle [début ; fin[si 2 paramètres sont renseignés.
`L=list(range(1,5))` va créer la liste [1 , 2 , 3 , 4] le premier terme sera `L[0]=1` et le dernier `L[3]=4`.
- Dans l'intervalle [début ; fin[mais de *pas* en *pas*, si les 3 paramètres sont renseignés.
`L=list(range(2,9,2))` va créer la liste [2 , 4 , 6 , 8] .

Exercice 2 : Deux type de répétitions

1- Recopier le programme Python et l'exécuter

```
# Programme 1
n=int(input('n='))
for k in range(0,11) :
    print(k,'*',n,'=',k*n)
```

Il faut bien respecter les tabulations

Que fait ce programme ?

2- Recopier le programme Python et l'exécuter

```
# Programme 2
import numpy.random as rd
import numpy as np
x=np.floor(rd.random()*10+1) #x est un nombre au hasard dans [1,10]
y=int(input('y='))
n=1
while y!= x :
    y=int(input('y='))
    n=n+1
print('Bravo vous avez gagné en',n,' coup(s)')
```

Il faut bien respecter les tabulations

Que fait ce programme

3- Expliquer la différence des deux programmes

Dans tous les exercices ,il faudra bien respecter les tabulations

Exercice 3 : Découverte de for

1- Écrire le script suivant et l'exécuter . Que se passe t-il ?

```
for i in range(1,11):
    print(i)
```

2- Écrire le script suivant et l'exécuter . Que se passe t-il ?

```
for i in range(1,11):
print(i)
```

La tabulation appelé indentation est fondamentale

Bilan 3: Indentation pour les blocs

- Dans le langage Python, on peut passer des lignes pour plus de clarté, ce qui n'est pas pris en compte lors de l'exécution du programme. Par contre, vous ne pouvez pas ajouter un espace en début de ligne comme bon vous semble, car cela a une signification.
- **Indentation** : On appelle *indentation* ce décalage d'un cran d'une ou de plusieurs lignes d'un programme. Elle permet de délimiter un bloc d'instructions dans une boucle ou lors d'une exécution conditionnelle.
- **Deux points " :"** La ligne précédent l'indentation se finit toujours par deux points. Quand vous appuyez sur la touche après avoir tapé « : », l'indentation est automatiquement effectuée en même temps que le passage à la ligne. On peut aussi appuyer sur la touche de tabulation *Tab*, à gauche de la touche "A" pour gérer l'indentation, mais les deux points : sont toujours nécessaires.

les **:** indiquent que les instructions commencent. Ne jamais les oublier

Bilan 3: `la commande for`

`for` *variable* in `range(debut,fin,pas)` **:**

bloc d'instructions qu'il ne faut pas oublier d'indenter

Réalise une boucle en faisant parcourir à la variable *variable* toutes les valeurs de `debut` à `fin - 1`.