

## TP N°1 : CONDITIONS ET BOUCLES

### I Booléen

Un booléen est une instruction qui a deux valeurs possibles True ou False : en général, cela permet de répondre à une question et donc de les utiliser comme condition.

Par exemple : tapez les instructions suivantes dans le shell :

```

1 x = 2 # affectation de la valeur 2 à la variable x
2 x > 2
3 x <= 2
4 x==2
5 not (x == 2)
6 2<x<5

```

Dans la suite, les conditions sont des booléens.

### II Instructions conditionnelles : if...elif...else ou pas

Trois formes d'utilisation possibles :

```

1 if condition1:
    {instruction1}

```

Si la condition1 est vraie (True), on exécute instruction1.

```

1 if condition1:
    {instruction1}
3 else:
    {instruction2}

```

Si la condition1 est vraie, on exécute instruction1, sinon (i.e la condition1 est fausse, False), on exécute instruction2.

```

1 if condition1:
    {instruction1}
3 elif condition2:
    {instruction2}
5 .
6 .
8 elif conditionn:
    {instructionn}
10 else:
    {instructionn+1}

```

Le programme sélectionne successivement les conditions suivant les mots elif, dès que l'une des conditions est vraie, la suite des instructions correspondantes est exécutée, dans le cas contraire, si else : est présente les instructions correspondantes sont exécutées et sinon, aucune instruction n'est exécutée.

- 1** Ecrire une fonction en Python de paramètres x, y qui renvoie le maximum des deux nombres sans utiliser la fonction max et si ce max est positif ou nul.
- 2** Ecrire une fonction en Python de paramètres x, y, z qui renvoie la valeur médiane des trois nombres.

### III Instructions itératives : for et while

- Syntaxe boucle for

```

1 # k prend toutes les valeurs de conteneur
2 # conteneur peut être une liste , un range ,
3 # une chaîne de caractère .
4 for k in conteneur:
    {instructions}
6 # les instructions dépendent ou non de k .

```

- 3** Testez le script suivant pour n = 3, n = 4 puis n = 5, qu'a-t-on programmé ?

```

1 n=int(input('n=?'))
2 p=1
3 for k in range(1,n+1):
4     p*=k
5 print(p)

```

• **Syntaxe boucle while :**

L'instruction `While` s'utilise pour répéter une ou plusieurs instructions tant qu'une condition est vraie : on ne connaît pas le nombre de répétitions nécessaires à l'avance. La syntaxe est la suivante :

```
1 while condition:
2     {instructions}
```

Si la condition est fausse dès la première évaluation, instructions n'est jamais effectuée.

Si la condition est toujours vraie , alors la répétition ne s'arrête jamais : la machine est plantée. Pour interrompre le programme, on presse simultanément sur `ctrl + l` ou on utilise le menu déroulant "shell+interrompre".

**4** Que fait le programme suivant? Testez le pour  $a=5$  puis  $a=10$  et enfin  $a=20$ .

```
1 a=float(input('entrez un réel a'))
2 S=0
3 k=0
4 while S<= a:
5     k+=1
6     S+=1/k
7 print(k)
```

## IV Exercices

**5** Pour tout  $n \geq 1$ , on pose  $S_n = \sum_{k=1}^n \frac{1}{k^3}$ .

Ecrire une fonction suite qui calcule  $S_n$ .

Tester ce programme pour  $n = 10, n = 100, n = 1000$ . Quelle conjecture pouvez vous faire?

**6** Ecrire une fonction suite\_exo6 qui permet de calculer  $u_n$  où  $(u_n)_{n \in \mathbb{N}}$  est définie par :

$$\begin{cases} u_0 = 0 \text{ et } u_1 = \frac{1}{2} \\ \forall n \in \mathbb{N}, u_{n+2} = \frac{1}{3}(1 + u_{n+1} + u_n^3). \end{cases}$$

**7** On considère la suite  $(u_n)_{n \geq 1}$  telle que  $u_1 \in ]0; \pi[$  et pour tout  $n \in \mathbb{N}^*$  :

$$u_{n+1} = \left(1 + \frac{1}{n}\right) \sin(u_n).$$

1) Représenter graphiquement à l'aide de Python  $u_n$  en fonction de  $n$  pour plusieurs valeurs de  $u_1$ , puis émettre une conjecture sur la monotonie de la suite  $(u_n)$ .

2) Emettre une conjecture sur la limite de :  $\sqrt{n} \times u_n$ .

**8** Ecrire une fonction moyensuite qui prend en argument une fonction  $f$  et un entier naturel  $n$  et qui renvoie  $\frac{f(0) + \dots + f(n)}{n+1}$ .

Tester avec les fonctions  $f_1 : x \mapsto \exp(1/(x+1))$  ,  $f_2 : x \mapsto \ln(2 + \frac{2}{x+2})$ ,  $f_3 : x \mapsto \sqrt{x} \exp(-x)$ . Que constate-t-on? Enoncer une propriété en lien avec la suite  $(f(n))_{n \in \mathbb{N}}$ .

Tester pour la fonction  $f_4 : x \mapsto (-1)^x$ . Que pouvez vous en déduire?

**9** Ecrire une fonction binome qui calcule  $\binom{n}{k}$ . On pourra commencer par montrer que :

$$\binom{n}{k} = \frac{\prod_{i=0}^{k-1} (n-i)}{\prod_{i=0}^{k-1} (i+1)}$$