

Introduction à la programmation – TP7 – 14 octobre 2020

Objectif : La boucle while en python

Exercice 1 – Saisie "tant que"

Écrire la fonction `saisieON` qui demande à l'utilisateur de répondre par oui ou non et qui recommence jusqu'à ce que l'utilisateur donne une réponse acceptable (dans un premier temps on acceptera uniquement les réponses "OUI" et "NON" on pourra en ajouter ensuite). Quand la réponse est "compréhensible" on renverra 1 pour OUI et 0 pour NON.

Exercice 2 – Saisie et moyenne

On veut refaire l'exercice de saisie de valeurs mais sans demander au préalable le nombre de valeurs à saisir. Le programme demande des valeurs à l'utilisateur jusqu'à ce que l'utilisateur tape la valeur "-1". Le programme affiche alors le nombre de valeurs saisies et leur moyenne.

```
premiere valeur ? 5
valeur suivante? -1 pour stopper 10
valeur suivante? -1 pour stopper 15
valeur suivante? -1 pour stopper 8
valeur suivante? -1 pour stopper 12
valeur suivante? -1 pour stopper -1
la moyenne des 5 valeurs saisies est 10
```

Exercice 3 – Espaces inutiles en début et fin de chaîne

a. Écrire une fonction qui supprime tous les espaces vides qui sont au début d'une chaîne de caractères (on renverra la chaîne obtenue). Exemple : `supprime(" truc bidule ")` renvoie "truc bidule".

b. Écrire une fonction qui supprime tous les espaces vides qui sont en fin d'une chaîne de caractères (on renverra la chaîne obtenue). Exemple : `supprime(" truc bidule ")` renvoie " truc bidule".

Exercice 4 – Test de lancers de dés –version 2

On veut écrire des programmes pour « tester le lancer de dés ou de pièces aléatoire » : On n’oubliera pas le `from random import randint` en début de fichier

1. Écrire la fonction `alea(n)` qui renvoie un entier entre 1 et n (inclus)
2. Deuxième programme
 - Écrire la fonction `aumoins(n)` qui lance une pièce autant de fois que nécessaire pour qu’on ait obtenu au moins n fois face et n fois pile (on représentera face par 1 et pile par 2). Cette fonction renverra le nombre d’essais nécessaires.
 - Écrire la fonction `main()` qui demande la valeur du paramètre n, lance l’expérience et affiche les résultats.

Exemple :

```
nombre voulu? 100
nombre de lancers nécessaires: 210
```

Exercice 5 – Exercice sur les mots

Une institutrice distribue des mots à ses élèves pour qu’ils apprennent à les écrire. . Chaque mot sera une chaîne de caractères (on ne prendra ni articles ni mots composés). Ces mots vont être mis dans des tuples pour avoir un ensemble de mots. Par exemple `t=('pomme', 'poire', 'ananas', 'banane', 'citron', 'carambole', 'kiwi', 'pastèque')`

On va reprendre l’exercice d’entraînement vu éventuellement au TP4 mais avec des modifications. On ajoute maintenant le fait qu’en cas d’erreur un enfant devra recommencer sur chaque mot jusqu’à ce qu’il l’écrive juste ou qu’il ait fait 5 tentatives.

Écrire la fonction `copie(m)` qui étant donnée un mot va afficher ce mot en demandant à l’enfant de le recopier. On juge la réponse et on affichera « bravo » ou alors « ce n’est pas juste » . Si ce n’est pas juste alors on demandera à l’enfant de recommencer jusqu’à ce que le mot soit bien écrit ou qu’il y ait eu 5 tentatives. Le programme doit renvoyer le nombre de fautes (entre 0 et 5).

```
copie('pomme')
recopie ce mot : pomme
pomme
bravo

copie('poire ')
recopie ce mot : poire
poirre
ATTENTION ce n'est pas juste
copie('poire ')
recopie ce mot : poires
ATTENTION ce n'est pas juste
copie('poire ')
recopie ce mot : poire
```

```
bravo
...
Tu as fait 1 faute en tout
bravo
```

Écrire ensuite une fonction qui étant donnée un tuple de mots va utiliser la fonction précédente pour faire travailler l'enfant sur chaque mot successivement. Quand on a utilisé tous les mots du tuple, on doit afficher le nombre de fautes commises par l'enfant et le féliciter s'il n'a fait aucune faute ou une seule faute ou lui faire une remarque s'il y a beaucoup de réponses étaient fausses.

Vous pourrez aussi améliorer le programme pour que le mot saisi par l'enfant soit mis en minuscule et débarrassé des espaces inutiles que l'enfant aurait pu mettre avant et après le mot (afin d'éviter qu'on ne compte faux un mot qui semble juste!) avant que l'ordinateur le teste.

Exercice 6 – divisibilité par 3

a. Écrire une fonction qui étant donnée une chaîne de caractères uniquement composée de chiffres, renvoie la somme de ces chiffres. exemple : `som("123123")` renverra 12 .

b. On va écrire un critère de divisibilité par 3 en utilisant le fait qu'un nombre est divisible par 3 si et seulement si la somme des chiffres le composant est divisible par 3. Pour cela on utilisera l'algorithme suivant :

soit `ch` la chaîne à étudier soit `s = som(ch)` si `s <= 9` faire le test (dans ce cas on pourra utiliser que seuls 3, 6 et 9 sont divisibles par 3) sinon recommencer

Écrire cette fonction de test.