Tutorat Python

Exercice 1 :

1) Demandez à l'utilisateur un nombre et dénombrer jusqu'à ce nombre de 1 en 1 (exemple : 1, 2, 3, 4, 5….)

2) Indiquer en plus si les nombres sont pairs ou impairs (exemple : 1 est impair, 2 est pair, 3 est impair….)

correction : 1to100.py

Exercice 2 :

Faire un petit jeu en générant aléatoirement un nombre entre 1 et 10 que l'utilisateur devra deviner. Le programme doit générer aléatoirement un nombre entre1 et 10, demander à l'utilisateur de deviner le nombre et indiquer à l'utilisateur s'il a gagné ou bien si le nombre est supérieur ou inférieur.

Aide : utiliser random de python, mettre *import random* en début de programme. La ligne suivante génère un nombre aléatoire entre 1 et 10 : random.randint(1,10)

correction : guess.py

Exercice 3 :

Refaire l'exercice 1 en utilisant la structure des listes.

Aide : liste = [] #créer une liste vide
liste = ['toto',5,'timmy'] #on peut aussi créer une liste avec des éléments
liste.append('test') #ajoutes l'élément 'test' à la fin (['toto',5,'timmy','test'])
len(liste) #retournes la taille ou le nombre d'éléments dans la liste, ici 4
liste[2] #retournes l'élément à l'indice 3 dans la liste, ici 'timmy'. Attention le premier élément d'une liste est toujours à l'indice 0, ici liste[0] retourneras 'toto'

correction : liste1to100.py

Exercice 4 :

Refaire l'exercice 3 en utilisant des fonctions.

correction : fun1to100.py

Exercice 5 :

1) Faire la suite de Fibonacci, chaque élément de la suite est égale à la somme des deux éléments précédents. Utiliser une liste pour stocker les éléments de la suite.
$U_{n+2} = U_{n+1} + U_n$
$U_0 = 0$; $U_1 = 1$

2) Améliorer le programme de sorte à ce que l'utilisateur puisse demander des éléments de la suite sans avoir à relancer le programme

correction : fibonacci.py

Exercice 6 :

Creer une banque contenant des comptes en banque (merci Captain Obvious). Un compte en banque est composé du nom du titulaire du compte et d'un montant. Il est conseillé de faire deux listes distinctes, une liste contenant les noms des titulaires et une autre contenant les montants.

1) Faites une fonction permettant d'ajouter un compte en banque.

2) Faites une fonction pour afficher l'ensemble de la banque, c'est à dire le nom de chaque titulaire et le montant de leur compte.

3) Faites une fonction permettant d'afficher la moyenne des comptes en banque.

4) Faites une fonction simulant une banqueroute. Vous prétextez avoir mal géré les actifs de votre banque et pour survivre vous devez prélever 10 % de chaque compte pour les virer sur votre propre compte, en gros vous les volez.

5) Faites un menu pour pouvoir accéder à toutes les fonctionnalités de votre programme

corretion : listebanque.py

Exercice 7 :

Un dictionnaire est une structure composé de clé et d'une valeur rattachée à cette clé (dictionnaire = {'ceci est une cle' : 'timmy', 2 :'les chiffres aussi peuvent etre des cles'}

Refaire l'exercice 4 en utilisant un dictionnaire à la place d'une liste.

Aide : dico = {} #creer un dictionnaire vide
len(dico) #retournes la taille du dictionnaire, soit le nombre de cle
dico['cle'] = valeur #ajoutes un element au dictionnaire avec la cle et la valeur

dico.keys() #retournes l'ensemble des cles du dictionnaire sous forme de liste

dico['cle'] #retournes la valeur de la cle renseigner entre crochet

correction : dic1to100.py

Exercice 8 :

1) Refaire l'exercice 6 avec une structure de dictionnaire.

2) Ajouter une fonction permettant d'afficher un seul compte en banque renseigner par le nom du titulaire

correction : dicbanque.py

Exercice 9 :

Vous partez à l'aventure dans le monde de Djoto, mais le professeur Chen a oublié de vous donner un pokedex.

1) Faites un pokedex pouvant contenir le nom d'un pokémon, son type et son nombre de PV. Vous devez pouvoir à tout moment ajouter un pokémon au cours de votre aventure.

2) Faites une fonction pour afficher tout les pokémons de votre pokédex

3) Faites une fonction pour afficher la moyenne des PVs de l'ensemble des pokémons de votre pokédex

4) Faites une fonction pour afficher les pokémons ayant le plus de PV, attention on veut afficher tout les pokémons avec cette valeur max, non pas un seul

5) Faites une fonction pour afficher tout les pokémons d'un même type

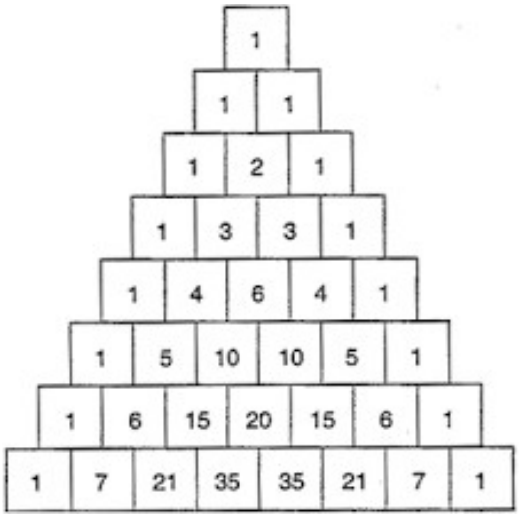
6) Faites une fonction pour faire monter d'un niveau et augmenter les PV d'un de vos pokémon

7) Faites un menu pour accéder à toutes les fonctionnalités de votre pokédex

correction : listepokedex.py

Exercice 10 :

Faites un triangle de pascal.



correction : trianglepascal.py