

# TD IA N° 1: Initiation au langage python

Redouane Ezzahir

2021

## Exercice 1

Taper dans l'interpréteur de commande Python (une par une) les instructions suivantes :

```
x=3.0 ; type(x) ; int(x) ; type(x) ; x=int(x) ; type(x) ;  
y=3 ; type(y) ; y=float(y) ; type(y) ;
```

- Que fait les fonctions type, int et float ?
- Quels types de variables avons nous mis en évidence ici ?
- Expliquer comment changer une variable de type flottant en une variable de type entier ? et réciproquement ?

## Exercice 2

Taper dans l'interpréteur de commande Python (une par une) les instructions suivantes :

```
A=['a', 'b', 'c'] ; type(A) ; B='abc' ; type(B) ;
```

Nous venons de mettre en évidence deux nouveaux types de variables : les listes et les chaînes de caractères. Expliquer comment changer une variable de type 'str' en une variable de type 'list' ? Vérifier. Réciproque ?

## Exercice 3

Taper dans l'interpréteur de commande Python les instructions suivantes :

```
A=1; A=bool(A); B = not A; B = A or B; A= A and B; type(A); type(B);  
Quel nouveau type de variable avons-nous mis en évidence ?
```

## Exercice 4

Compléter le code suivant qui converti une liste en un chaîne de caractères

```
#####  
#Programme Python pour convertir une liste en chaîne  
#Fonction de conversion  
def listToString(string):  
    # initialise une chaîne vide  
    str1 = ''  
    # traverser la chaîne  
    for ch in string:  
        #votre code ici  
    return str1  
#-----  
# code de test  
s = ['L'intelligence', 'est', 'la faculté à connaître', 'à comprendre', 'et à raisonner']  
print(listToString(s))  
#alternative  
print(' '.join(s))  
#####
```

## Exercice 5

Saisir un nom et un age en utilisant l'instruction `input()`, les afficher, et refaire la saisie du nom, mais avec l'instruction `raw_input()`. L'afficher. Enfin, utilisez la bonne pratique : recommencez l'exercice en transtypant les saisies effectuées avec l'instruction `raw_input()`

```
# ----- programme principal -----
## instruction input :
nom = input("ton nom ? ") # essayer raw_input("ton nom ? ")
age = input("age ? ")
age = float(age)
print("\t Nom :", nom, "\t Age :", age)
#
# ----- bonnes pratiques : -----
nom = raw_input("ton nom ? ") # pour une chaine
age = float(input("age ? ")) # sinon : transtyper explicitement
print("{}\n\t Nom : {} \t Age : {}".format("-"*40, nom, age))
#-----
```

## Exercice 6 : Générateur de QCM d'IA

Vous allez écrire un programme qui génère des questions à choix multiples (QCM) d'intelligence artificielle. Le programme doit permettre le stockage de toutes les questions et réponses "Quiz" que nous verrons dans le cours d'IA. Vous utiliserez ce programme pour tester votre progression d'apprentissage.

1. Créer le fichier `qcm.ia.py` et importer la methode `shuffle` du module `random` :

```
import copy
from random import shuffle
```

2. Initialiser une variable `QR` de type liste qui stock les questions et leurs réponses :  
*#liste de questions et leurs réponses*  
`QR = []`
3. Définir une fonction `nouvelleQR` (`question`, `choix`, `rep`) qui transforme le tuple donné en parametre en une liste

```
def nouvelleQR (question, choix, rep) :
    # question : la question
    # choix: les choix multiples sous forme de liste
    # rep: liste de boolean de même taille que de la liste choix
    ####votre code ici ####
    #####
```

4. Définir la fonction `ajouterQR` (`qr`) qui ajoute à la queue de la liste `QR` globale une question et ses réponses donnée en parametre

```
def ajouterQR (qr) :
    global QR
    ####votre code ici ####
    #####
```

5. Définir la fonction `ordreAleatoire()` qui renvoie une liste contenant les indices des questions en `QR` mais classés aléatoirement.

```
def ordreAleatoire() :
    global QR
    n = len(QR)
    ordre = range(n)
    ####votre code ici ####
    #####
```

- Créer un fichier QR.py dans lequel nous enregistrons les questions/réponses sous forme d'appelles aux méthodes nouvelleQR et ajouterQR . Exemple la question suivante sera ajouter à la base des QR par le code en dessous.

Quand le terme « Intelligence Artificielle » a-t-il été fondé ?

A : années 1930, B : années 1960, C : années 1990, D : années 2000

```

1  print ( "Chargement des  Qestions  et Reponses ...")
2
3  # ajoute de la première QR
4  q = "Quand le terme « Intelligence Artificielle » a-t-il été fondé ?"
5  choix = ['années 1930', 'années 1960', 'années 1990', 'années 2000']
6  rep = [false, true, false, false]
7  ajouterQR(nouvelleQR(q, choix, rep) )

```

- Faire appelle au script QR.py dans le script principal qcm.ia.py.  
**NB :** Pour exécuter un script Python dans l'espace de noms du script appelant, nous utilisons la ligne suivante : `exec(open("./autre_script.py").read())`
- Compléter le programme suivant qui exécute la boucle pour générer des questions et lire les réponses. Le score sera donné en pourcentage. L'utilisateur peut interrompre la boucle en répondant par le caractère 'q'.

```

n = len(QR)
while True:    # infinite loop
    n = int(input("Nombre de questions <= {} : ".format(len(QR)) ))
    if  n > 0  and n <= len(QR) :
        break  # stops the loop when n is valid

#indices des questions
indices = range( len(QR))
#mélanger les indices
shuffle(indices)
score=0.0
#afficher et lire les n premières questions obtenues après le mélange
for i in range(n):
    idx = indices[i]
    qcm = QR[idx]
    ###votre code ici ##
    ## si la reponse est incorrect afficher "Incorrect! "  et la bonne réponse##
    ##si non afficher "Très bien, continuez (o/n)?"

#affichage du score
print(" votre score={} ".format(score/n))

```

## Exercice 6.9 (suite)

Ajouter les questions suivantes à la base des QR :

- Quelle technologie clé se cache derrière l'intelligence artificielle ?  
A : Apprentissage automatique    B : Robotique,    C : Pile électrique,    D : Chaîne de blocs
- Comment l'intelligence artificielle est-elle développée davantage et comment les erreurs dans le système d'IA sont-elles corrigées ?  
A : Grâce à une connexion Internet plus rapide.  
B : par apprentissage : l'utilisateur donne un retour et apporte des corrections à l'IA  
C : Un développement ultérieur n'est pas possible avec ces systèmes, car ce sont des programmes prêts à l'emploi.  
D : Grâce à des capacités de stockage plus élevées des ordinateurs

## Exercice 7 : Piles, Files d'attentes et Files d'attentes prioritaires

Les piles, les files d'attente et les files d'attente prioritaires sont toutes des structures de données très basiques et largement utilisées en IA. Vu que la plupart des structures de données de base sont déjà implémentées en Python, nous ne devons pas dupliquer les roues ; Nous devons choisir ces structures de données qui répondront à nos besoins. la classe `collections.deque` est une liste doublement chaînée qui peut être utilisée pour implémenter des Piles et des Files d'attentes dans le cas d'un seul thread.

1. créer le script python `util.py` en important `deque` du module `collections`

```
from collections import deque
```

2. En étendant la classe `deque`, implémenter une classe pile (FILO)

```
class pile(deque):
    def __init__(self):
        deque.__init__(self)

    def empiler(self, item):
        ###votre code ici ###

    def depiler(self):
        ###votre code ici ###
```

Tester votre implementation :

```
p.empiler(0)
p.empiler(5)
p.empiler(10)
p.empiler(15)
print(p.depiler())
print(p.depiler())
```

3. En étendant la classe `deque`, implémenter une classe file d'attente (FIFO)

```
class file(deque):
    def __init__(self):
        deque.__init__(self)

    def emfiler(self, item):
        ###votre code ici ###

    def defiler(self):
        ###votre code ici ###
```

Tester votre implementation :

```
p.emfiler(0)
p.emfiler(5)
p.emfiler(10)
p.emfiler(15)
print(p.defiler())
print(p.defiler())
```

4. la classe `queue.PriorityQueue` réalise une file d'attente prioritaire. En suivant le même principe des deux questions précédentes implémenter une classe `fileprioritaire` qui étende la classe `queue.PriorityQueue`.
5. Lisez la documentation du module `heapq` et la classe `queue.LifoQueue`

## Exercice 8 : QCM

1. Qu'est-ce que l'intelligence artificielle ?  
A : Mettre votre intelligence dans l'ordinateur  
B : Programmer avec votre propre intelligence  
C : Rendre une machine intelligente  
D : Jouer à un jeu
2. Qui a inventé la première Intelligence Artificielle ?  
A : Arthur Samule  
B : James Slagle  
C : Jhon McCarthy  
D : EF Morue

3. Lequel des éléments suivants n'est pas l'avantage de l'IA ?  
A : Grande vitesse      B : Menaçante      C : Précision      D : Coût élevé
4. Les caractéristiques du système informatique capable de penser, de raisonner et d'apprendre sont connues  
A : intelligence artificielle      B : intelligence humaine      D : intelligence virtuelle
5. Le succès d'un comportement intelligent d'un système peut être mesuré avec un :  
A : Test du système      B : Test de Turing      C : Test intelligent      D : Test de machine
6. Laquelle des applications suivantes n'est pas de l'IA  
A : Anti-virus,      B : Tesla      C : Siri      D : Netflix
7. Lequel des domaines suivants ne peut pas contribuer à construire un système intelligent ?  
A : Science des neurones      B : Mathématiques      C : L'informatique      D : Géologie
8. En quelle année le terme Intelligence Artificielle a été inventé ?  
A : 1956      B : 1957      C : 1965      D : 1959
9. Si un robot peut modifier sa propre trajectoire en réponse à des conditions externes, il est considéré comme :  
A : intelligente      B : Extra ordinaire      C : Testeur de Turing      D : Connaissance