

EMD Remplacement Langage Evolué | SOLUTION

EXERCICE 1 : QCM (10 points)

Q1 : Que va renvoyer la commande suivante :

```
>>> type ( " {1,2} " )
```

- <class 'list'>
- <class 'tuple'>
- <class 'dict'>
- <class 'str'>

Q2 : Que va renvoyer la commande suivante :

```
>>> type ( [ ("1:2" , "4:8") , (1,4) ] )
```

- <class 'list'>
- <class 'tuple'>
- <class 'dict'>
- <class 'str'>

Q3 : Soit :

```
>>> a = [ [1 , 2 , 3] , " { 1:2 } , [ 1 , 2 ] " ]
>>> len(a)
```

Que va afficher le code suivant : len(a)

2

Q4 : L'expression

"fruits" in ("kiwi","banane","pomme") retourne:

- <class 'tuple'>
- True
- False
- str

Q5 : Les séquences immuables sont...:

- Chaîne
- Tuple
- Liste
- Dictionnaire

Q6 : Soit la variable fruits initialisée avec le tuple ("kiwi","pomme","mangue"). L'instruction fruits[0] = "poire" est elle correcte ? :

- Oui
- Non
- Oui sous certaines conditions

Q7: Soit la variable fruits initialisée avec le tuple

```
("kiwi", "pomme", "mangue", "poire",
 "orange", "figue").
```

L'instruction fruits[1:3] retourne... :

- ()
- ("pomme","poire")
- ("pomme","mangue")
- ("pomme","mangue","poire")

Q8 : Indiquez ce que va afficher le code suivant si l'utilisateur tape 100 en réponse à l'instruction « input »:

```
try:
    x = float( input("Donne une valeur") )
except:
    print ("Erreur")
else:
    print (x)
```

100.0

Q9 : Soit une liste initialisée comme suit :

```
>>> listeMots = ['Bejaia', 'est', 'une', 'ville.']
```

Quelle est la commande python permettant de créer, à partir de la variable « listeMots », une nouvelle variable « phrase » contenant une chaîne composée des éléments de la variable « listeMots » séparés par des espaces.

Phrase = " ".join(listeMots)

Q10 : Soit les commandes suivantes :

```
>>> pays = dict()
>>> pays["Maghreb"] = [ "Algérie", "Tunisie", "Maroc"]
>>> pays["Europe"] = [ "France", "Belgique", "Italie"]
>>> pays["Europe"] = [ "Allemagne", "Turquie"]
>>> print(len(pays))
```

Que va afficher la dernière commande ?

2

Q11 : Que va afficher le code suivant ?:

```
def afficher() :
    print ( " Bonjour " )
    print ( " monsieur " )
```

monsieur

Q12 : Soit les commandes suivantes :

```
>>> Club = ['JSK', 'MOB', 'JSMB']
>>> print ( Club[-1] )
```

Que va afficher la dernière commande ?

JSMB

Q13 : Que va afficher le code suivant ?:

```
def afficher() :
    print ( " Bonjour " )
    return
    print ( " monsieur " )
afficher()
```

bonjour

Q14 : Que va afficher le code suivant ?:

```
>>> x = "10"
>>> y = 20
>>> print( float( str( int(x)+y)*2 ) )
```

3030.0

Q15 : Que va afficher le programme suivant :

```
>>> s = "*****Béjaia ????"
>>> r = s.strip( "*" )
>>> r = r.strip( " ?" )
>>> print ( r )
```

Béjaia

Q16 : Que va afficher le programme suivant :

```
>>> v = 20
>>> def f ( x , y = 1):
        x = x - y
        return x+v
>>> print ( f ( v ) )
```

39

Q17 : Que va afficher le programme suivant :

```
>>> v = 20
>>> def f ( x, y = 1):
        x = x - y
        return x+v
>>> print ( f ( 5, 6 ) )
```

19

Q18 : Que va afficher le programme suivant :

```
>>> v = 6
>>> def f ( x, y = 1):
        x = x + y
        return x + str ( v )
>>> print ( f ( " Ville ", " Bejaia " ) )
```

Ville Bejaia 6

Q19 : Que va afficher le programme suivant :

```
>>> D = dict()
>>> def f(x):
        For i in range(3) :
            D[i] = i*2
>>> f(10)
>>> for v in D.values() :
        print(v)
```

0

2

4

Q20 : Que va afficher le programme suivant :

```
>>> def f ( x , y ) :
        return x + y
>>> print ( f ( [1, 2, 3], [0] ) )
```

[1, 2, 3, 0]

EXERCICE 2 (2 points) : Écrire une fonction "**compterMots**" ayant comme argument une chaîne de caractères représentant une phrase et qui renvoie un dictionnaire contenant la fréquence (nombre d'apparitions) de tous les mots de la chaîne entrée.

Indication : On suppose que la phrase est correctement construite.

Solution

```
def compterMots(phrase):
    liste1 = phrase.split()
    liste2 = phrase.split()
    for e in liste2:
        if e not in d.keys():
            d[e]=1
        else :
            d[e]=d[e]+1
    return d
```

EXERCICE 3 (2 points) : Écrire un programme qui lit un texte au clavier et qui affiche uniquement les chiffres saisis.

Solution

```
texte = input("donnez un texte")
chiffres = "0123456789"
for e in texte:
    if e in chiffres:
        print(e)
```

EXERCICE 3 (6 points) : On souhaite écrire un logiciel qui évalue de façon automatisée une rédaction d'un élève. A cet effet, on suppose qu'on a :

1. Une rédaction d'un élève consignée dans un fichier nommé « **redaction.txt** ». Ce fichier n'est rien d'autre qu'un texte.
2. Un second fichier, nommé « **conceptsAttendus.txt** », qui contient une liste de concepts qui devraient apparaître dans une bonne rédaction. On supposera que chaque ligne contient un concept (un concept est composé d'un ou de plusieurs mots).

Le logiciel que l'on souhaite écrire doit attribuer une note à la rédaction. Cette note est calculée comme suit : **note = (nbConceptsTrouvé / nbConceptsAttendus)*20**

avec :

- *nbConceptsTrouvé* : Nombre de concepts clés trouvés observés dans la rédaction de l'élève.
- *nbConceptsAttendus* : nombre de concepts clés attendus (se trouvant dans le fichier « **conceptsClés.txt** »)

```
f = open("conceptsAttendus.txt","r")
conceptsAttendus = f.read()
conceptsAttendus = conceptsAttendus.split("\n")
f.close()

f = open("redaction.txt","r")
redaction = f.read()
redaction = redaction.split()
f.close()

nbConceptsAttendus = len(conceptsAttendus)

nbConceptsTrouvé = 0
for e in conceptsAttendus:
    if e in redaction:
        nbConceptsTrouvé = nbConceptsTrouvé+1

note = (nbConceptsTrouvé/nbConceptsAttendus)*20

print("voici la rédaction de l'élève : ")
print(" ".join(redaction))
print("voici la note trouvée par l'ordinateur : ", note)
```

Récupération des concepts

Récupération de la rédaction

Calcul du nombre de concepts

Calcul du nombre de concepts observés sur la rédaction

Calcul de la note sur 20

Affichage du résultat