Activités Python 1ère NSI

Lycée Dumézil

2021-06-12

Table des matières

Python Première

— Vous trouverez ici des activités qui vont vous permettre de réactiver et renforcer vos compétences en Python.
— Comme tout ce qu'on apprend, on l'oublie si on ne pratique pas assez souvent. Alors revenez souvent ici ;-)

Chapitre 1

Types de Variables

I. Activité 1 : float et int

Dans une console Python, saisie les entrées suivantes :

```
nombre_entier = 3
nombre_flottant = 3.14
type(nombre_entier)
type(nombre_flottant)
double = nombre_entier * 2
somme = nombre_entier + nombre_flottant
double
somme
```

1. Quel est le type de la variable `double`. Vérifie en utilisant l'instruction `type(double)`.
2. Quel est le type de la variable `somme`.
3. Quelle est l'erreur commise par Python lors du calcul de la somme ? Propose une explication.

II. Activité 2 : Divisions et modulo

Il existe deux opérations divisions en Python :

1. La division entière `//`
2. La division habituelle : `/`
3. Saisie dans la console Python les calculs suivants :

```
16 / 5
```

```
## 3.2
```

```
16 // 5
```

```
## 3
```

```
16 % 5
```

```
## 1
```

2. Réécrire le code précédent mais en utilisant des variables pour stocker 16 et 5 ainsi que les résultats des calculs.
3. Quelle différence y a-t-il entre les deux divisions ?
4. Quelle opération permet d'obtenir le quotient entier de la division ?

L'opération `%` s'appelle modulo, elle retourne le reste de la division. Puisque $16 = 3 \times 5 + 1$ on a `16//5 = 3` et `16%5 = 1`.

4. Vérifie dans la console que le modulo d'un entier pair vaut toujours 0 et que le modulo d'un entier impaire vaut 1. C'est un moyen de tester rapidement si un entier est pair ou non.
5. Un sac content 257 popcorns, calcule avec la console Python comment répartir les popcorns entre 3 personnes. Les popcorns restant seront pour le chien. Combien Médor reçoit-il de popcorn ?

III. Activité 3 : Chaînes

Objectifs : type chaine, concaténation, conversion de type, affichage

Les chaînes sont un type de variable pouvant contenir : un caractère, un mot, un texte.

```
ma_chaine = "Hello world !"
```

1. Vérifie avec l'instruction `type` le type de la variable `ma_chaine`.

```
a = "Je"
b = " vois"
nb = 3
c = " pommes dans cet arbre."
```

2. Ajoute les chaînes a et b. Quel résultat obtiens-tu ? On dit que les chaînes a et b ont été `concaténées`.
3. Stocke le résultat de `a+b` dans une variable `chaine_1`.
4. Essaye d'ajouter `chaine_1` à `nb`. Relève le message d'erreur qui apparaît. Explique cette erreur.

L'opération `+` n'est pas la même entre 2 chaînes et 2 nombres même si on utilise le même symbole + dans le code.

5. Saisie le code suivant :

```
chaine_nb = str(nb)
type(chaine_nb)
```

Explique à quoi sert `str()`

6. Complète le programme pour que `message` contienne la chaîne : "Je vois 3 pommes dans cet arbre"

```
a = "Je vois "
nb = 3
c = " pommes dans cet arbre"
nombre = ...
message = a + nombre + c
```

IV. Activité 4 : Affichage de chaines

Découverte

Jusqu'ici pour connaître le contenu d'une variable, on a tappé son nom dans la console puis appuyer sur entrée. On peut aussi utiliser une instruction pour afficher dans la console le contenu d'une variable ou une chaine de caractère.

1. Saisie dans la console python le code suivant :

```
c1 = "bonjour"
c2 = ' le monde !'
print('hello')
print(c1 + c2)
```

2. Détaille ligne à ligne l'exécution de ce script.
3. Modifie la dernière ligne du script afin d'affiche : `bonjour bonjour le monde !`

fstrings sont des chaînes de caractère plus complexes mais très utiles pour afficher des messages contenant des informations variables.

1. Saisie dans la console python le code suivant :

```
prof = "M.DENIS"
matiere = "NSI"
nb_eleve = 19
msg = f"{prof} enseigne la {matiere} a {nb_eleve} élèves de première."
print(msg)
```

2. Détaille ligne à ligne l'exécution de ce script.
3. Modifie ce code pour afficher M.MARTIN enseigne la Mathématique a 35 élèves de première.
4. Ajoute une nouvelle variable `niveau` qui contient la chaine "terminale" afin d'afficher : `M.MARTIN enseigne la Mathématique a 35 élèves de terminale.`

Mise en forme de l'affichage des nombres

Les fstrings permettent de mettre en forme l'affichage des nombres. Voyons la syntaxe :

— `f ' <texte> { <expression/variable> : <format> } <texte> ... '`
— le format s'écrit selon : `[alignement][signe][largeur][groupage][.précision][type]`

1. Saisie le code suivant :

```
n1 = 3.1415926
n2 = 6358
n3 = - 123.0034
print(f'nombre 1 {n1 : <10_.2f} nombre 2 {n2 : <10_.3e}')
```

2. En utilisant la documentation ci-dessous, détaille et justifie l'affichage du programme ci-dessus.

Documentation

— alignement : détermine ou le nombre est aligné dans sa zone
  — "$>$" aligne à droite
  — "$<$" aligne à gauche
  — "$\hat{}$" centré
  — "=" aligne le signe à gauche et le nombre à droite
— signe : détermine l'affichage du signe
  — "+" indique que le signe + doit être affiché ainsi que le -

  - “-“ indique que le signe - doit être affiché (par defaut)
  - “ “ n’affiche pas le + mais insére un espace à la place
- largeur : détermine la place qui doit être réservée pour l’affichage du nombre
- groupage : détermine le symbole de séparation tous les 3 chiffres
  - “ _”
  - “,”
- précision : détermine le nombre de chiffres après la virgule
- type : détermine le mode d’affichage
  - “e” ou “E” notation scientifique
  - “f” ou “F” affichage classique

3. Modifie le code précédent pour afficher dans un espace de 10 caractères, aligné à gauche, n3 en écriture scientifique avec 3 chiffres derrière la virgule.
4. On veut calculer le poids d’un humain de `m=50` kg sur la Terre et sur la Lune.

$$F_{Terre} = G \times \frac{m \times M_{Terre}}{r^2}$$

   Complète le programme suivant pour qu’il affiche

- `Le poids d un homme de 50 kg vaut sur la Terre 490.0      N.`
- `Le poids d un homme de 50 kg vaut sur la Lune 6.2        N.`

```
G = 6.67e-11
M_terre = 5.98e24
M_lune = 7.6e22
m = 50
r_T = 6380e3 #rayon de la Terre
r_L = 1737e3 #rayon de la Lune
F_T = G*m*M_terre/r_T**2 #Calcul du poids sur Terre
F_L = .. à compléter ...

print(f'Le poids d un homme de { à compléter} kg vaut sur la Terre {F_T : <10...compléter...} N.')
print(...Compléter...)
```

V. Corrigés des Activités

V.1. Activité 1 corrigé

1. double est une variable de type flottant : python renvoie `float` après exécution de `type(double)` dans la console.
2. Lorsque l'on addition un float avec un int le résultat doit pouvoir contenir la partir décimal du float. C'est pourquoi somme est du type float.
3. Python affiche que 6 + 3.14 = 6.140000000000001, il commet une erreur de 0.000000000000001. C'est peu mais c'est une erreur qui risque de se cumuler avec d'autres par la suite. Seul les nombres entiers sont exacts en python.

V.2. Activité 2 : Divisions et modulo correction

```
16 / 5
```

```
## 3.2
```

```
16 // 5
```

```
## 3
```

```
16 % 5
```

```
## 1
```

```
a = 16
b = 5
quotient = a / b
quotient_entier = a // b
reste = a % b
```

2. `/` calcul le quotient décimal
3. `//` renvoie la valeur entière du quotient.
4.

```
# NB de popcorns pour chaque personne
popcorns = 257
nb_personnes = 3
popcorns // nb_personnes
# Pour Médor
```

```
## 85
```

```
popcorns % nb_personnes
```

```
## 2
```

V.3. Activité 3 : Chaînes correction

Objectifs : type chaine, concaténation, conversion de type, affichage

Les chaînes sont un type de variable pouvant contenir : un caractère, un mot, un texte.

1. Vérifie avec l'instruction `type` le type de la variable `ma_chaine`.

```
ma_chaine = "Hello world !"
type(ma_chaine)
```

```
## <class 'str'>
```

2. Ajoute les chaînes a et b. Quel résultat obtiens-tu ? On dit que les chaînes a et b ont été `concaténées`.

```
a = "Je"
b = " vois"
a+b
```

```
## 'Je vois'
```

3. Stocke le résultat de `a+b` dans une variable `chaine_1`.

```
a = "Je"
b = " vois"
nb = 3
chaine_1 = a + b
chaine_1
```

```
## 'Je vois'
```

4. Essaye d'ajouter `chaine_1` à `nb`. Relève le message d'erreur qui apparaît. Explique cette erreur.

```
a = "Je"
b = " vois"
nb = 3
chaine_1 = a + b
chaine_1 + nb
```

Explication `chaine_1` et `nb` sont deux variables qui contiennent des informations de type différents. On ne peut additionner que des données de même type. On peut additionner deux entiers, un flottant avec un entier ou encore deux chaines mais pas une chaine et un entier.

5. Saisie le code suivant :

```
chaine_nb = str(nb)
type(chaine_nb)
```

```
## <class 'str'>
```

`str()` permet de convertir le nombre `nb` en une donnée de type chaine de caractère. La valeur 3 contenue dans nb est transformée en **chaine de caractère** "3". Il est alors possible de concaténé "3" avec une autre chaine.

6. Complète le programme pour que `message` contienne la chaîne : "Je vois 3 pommes dans cet arbre"

```
a = "Je vois "
nb = 3
c = " pommes dans cet arbre"
nombre = str(nb)
message = a + nombre + c
message
```

```
## 'Je vois 3 pommes dans cet arbre'
```

V.4. Activité 4 : fstring correction

Découverte

2.

```
c1 = "bonjour" #affectation de "bonjour" à la variable c1
c2 = ' le monde !' #affectation de " le monde !" à la variable c2
print('hello') #affiche le message hello
```

```
## hello
```

```
print(c1 + c2) # affiche la valeur de c1 + c2 c est a dire bonjour le monde
```

```
## bonjour le monde !
```

3. On ajoute " " entre deux c1 pour obtenir l'espace entre les deux bonjour.

```
print(c1+ " " + c1 + c2)
```

```
## bonjour bonjour le monde !
```

fstrings

1. msg est une `fstring`. Entre accolade on trouve le nom des variables dont on veut afficher le contenu.

```
prof = "M.DENIS"
matiere = "NSI"
nb_eleve = 19
msg = f"{prof} enseigne la {matiere} a {nb_eleve} élèves de première."
print(msg)
```

```
## M.DENIS enseigne la NSI a 19 élèves de première.
```

2. Détaille ligne à ligne l'exécution de ce script.
3. Modifie ce code pour afficher M.MARTIN enseigne la Mathématique a 35 élèves de première.

```
prof = "M.MARTN"
matiere = "Mathématique"
nb_eleve = 35
niveau = "terminale"
msg = f"{prof} enseigne la {matiere} a {nb_eleve} élèves de première."
print(msg)
```

```
## M.MARTN enseigne la Mathématique a 35 élèves de première.
```

4. Ajoute une nouvelle variable `niveau` qui contient la chaine "terminale" afin d'afficher : `M.MARTIN enseigne la Mathématique a 35 élèves de terminale.`

```
prof = "M.MARTN"
matiere = "Mathématique"
nb_eleve = 35
niveau = "terminale"
msg = f"{prof} enseigne la {matiere} a {nb_eleve} élèves de {niveau}."
print(msg)
```

```
## M.MARTN enseigne la Mathématique a 35 élèves de terminale.
```

Mise en forme de l'affichage des nombres

1. Saisie le code suivant :

```
n1 = 3.1415926
n2 = 6358
n3 = - 123.0034
print(f'nombre 1 {n1 : <10_.2f} nombre 2 {n2 : <10_.3e}')
```

```
## nombre 1 3.14       nombre 2 6.358e+03
```

2.

— n1 affiche avec alignement à gauche `<` et sur `10` caractères le caractère `_` pour séparer les milliers si besoin et `.` comme virgule. On affiche `2` chiffres après la virgule en notation classique (décimale).
— n2 idem que n1 mais en notation scientifique avec 3 chiffres derrière la virgule.

3. Modifie le code précédent pour afficher dans un espace de 10 caractères, aligné à gauche, n3 en écriture scientifique avec 3 chiffres derrière la virgule.

```
print(f"nombre 3 {n3 : <10_.3e}")
```

```
## nombre 3 -1.230e+02
```

4.

```
G = 6.67e-11
M_terre = 5.98e24
M_lune = 7.6e22
m = 50
r = 6380e3
F_T = G*m*M_terre/r**2
F_L = G*m*M_lune/r**2

print(f'Le poids d un homme de {m} kg vaut sur la Terre {F_T : <10_.1f} N.')
print(f'Le poids d un homme de {m} kg vaut sur la Lune {F_L : <10_.1f} N.')
```

Chapitre 2

Structures Conditionnelles

I. Activité 1 : Expression booléenne

— Une variable booléenne contient soit la valeur `True` soit la valeur `False`.
— Une expression booléenne contient au moins un opérateur booléen sa valeur est vraie ou fausse.

1. Relever dans ce programme les affectations et les expressions booléennes.

```
majeur = True
age = 18
age == 18
age < 18
age != 12
(majeur == True) or (age == 18)
age = 16
(majeur == True) and (age >= 18)
not majeur
age != 18
```

2. Justifier la valeur de chaque expression booléenne.
3. En modifiant les valeurs de `a` et `b` écrire la table de vérité de l'opérateur booléen `or`

```
a = True
b = False
a or b
```

3. Complète le programme suivant afin que les expressions booléennes soient vraies :

```
a = 3
b = False
c = 5
(a ... c) or b
not ( a ... 3)
(... b) and ( a == ...)
```

4. L'opérateur booléen `or` est un ou inclusif. Dans l'expression `a or b` si a et b sont vraies alors l'expression est aussi vraie. On veut écrire une expression booléenne qui se comporte comme un ou exclusif c'est à dire en valant faux si a et b sont vraies. Par exemple : dans l'expression "fromage ou déssert ?" on ne valide pas le choix fromage ET déssert. Compléter le code ci-dessous.

```
fromage = False
dessert = False
# Expression booléeenne à écrire
```

5. A l'aide de votre programme écrire la table de vérité de cette opérateur OU-Exclusif

II. Activité 2 : Si ... Alors

L'instruction IF permet d'exécuter un bloc d'insctrtions à la condition qu'une expression booléenne possède la valeur vrai. Par exemple :

```
expression_1 = True

if expression_1 :
  print("Bloc 1")
  expression_1 = False

print("Bloc 2")

if expression_1 :
  print("Bloc 3")

print("FIN")
```

1. Recopie et test le programme ci-dessus.
2. Explique et justifie ligne à ligne son exécution.
3. Modifie le code suivant afin que le programme affiche "Accès autorisé" si l'age est supérieur ou égal à 18.

```
age = 15
if .... :
  print("Accès autorisé")
```

4. Corriger les erreurs dans le programme suivant :

```
admin = True
age = 21
if (admin = True) and (age > 18)
print("Connecté en tant qu' ADMIN !")
```

5. Un hôtel loue des chambres avec un tarif intéressant si on arrive pas trop tard. Plus on arrive tôt et moins on paye.

— Le prix de la chambre est de 10 pièces à midi, et augmente de 5 pièces chaque heure après midi. Il est donc de 15 pièces à 13h, etc. Il ne peut cependant pas dépasser 53 pièces.
— Dans votre programme, utilisez une variable nommée `heure` qui contient un entier représentant l'heure d'arrivée. `heure` vaut 0 si on arrive à midi, 1 si on arrive à 1h de l'après-midi, etc, 12 si on arrive à minuit.
— Votre programme devra afficher le prix à payer correspondant à l'heure d'arrivée.

III. Activité 3 : if.. elif..else

1. Saisir et commenter en détail le programme suivant :

```
note = 11
if note >=10:
  print('Bonne note')
else:
  print('Mauvaise note')
```

2. Modifier le programme pour qu'il affiche "Mauvaise note".
3. Corriger l'erreur dans le programme suivant :

```
a = 12
if a >= 18:
  print('majeur')
else a<18:
  print('mineur')
```

4. Compléter le programme suivant attribue une mention au bac.

— Inférieur à 10 : Recalé
— Entre 10 et 12 : Passable
— Entre 12 et 14 : Assez-bien
— Entre 14 et 16 : Bien
— 16 et plus : Très bien

```
note = 15
if note >= 16:
  print('Très bien')
elif note >= 14:
  print('Bien')
elif ... :
  print('Assez-bien')
elif note >= 10:
  ....
else:
  ....
```

5. Modifier le programme précédent en inversant l'odre des deux premiers tests conditionnels. Que se passe-t-il ? Justifie.
6. Ecrire un programme qui affiche "pair" si la valeur de la variable `n` est paire et "impair" dans les autres cas. Pour savoir si un nombre est pair, on utilise le fait que le reste de la division par 2 de ce nombre vaut 0. On utilisera donc l'opérateur `%`.

IV. Activité 4 : Combinaisons de if, elif, else

1. Prédire et expliquer la valeur de `b` après exécution du programme suivant :

```
a=2
b=0
if a<0:
    b=1
elif a>0 and a<5:
    b=2
else:
    b=3
```

2. Commenter l'exécution du programme suivant :

```
x = -3
if x < 0:
  if x % 2 == 0:
    print('pair et négatif')
  else:
    print('impair et négatif')
else:
    print("pair et c'est tout ce qui m'intéresse")
```

3. Modifier le programme afin qu'il affiche pair et positif ou impair et positif si c'est le cas.

4. Le programme suivant est buggé. On souhaite que lorsque l'utilisateur a plus de 18 ans et possède son code et le permis le programme affiche "vous pouvez conduire". Modifiez légèrement le programme afin de fixer cette erreur.

```
age = 19
code = True
permis = True

if age >= 14:
    print('vous pouvez apprendre à conduire un scooter')
elif code == True and age >=18:
    if permis == True:
        print('vous pouvez conduire une voiture')
    else:
        print('vous pouvez apprendre à conduire une voiture')
```

```
## vous pouvez apprendre à conduire un scooter
```

4. Ecrire un programme qui affiche la phrase "il va faire très froid / froid / frais / bon / un peu chaud / chaud / très chaud." selon la valeur de la variable `t` représentant la température maximale prévue. Vous choisirez les seuils entre chaque niveau de température.
5. Une année est bissextile si elle est divisible par 4. Cependant les siècles ne sont pas bissextiles, sauf les multiples de 400. Ecrire un programme qui affiche si une année est bisextile. Vérifiez que 2100 n'est pas bissextile contrairement à 2012.
6. Plus loin : Ecrire une fonction qui reçoit une année et qui renvoie l'année si elle est bissextile et la chaine " " si elle n'est pas bissextile. En utilisant cette fonction écrire un script qui affiche toutes les années bissextiles entre 2000 et 2104.

V. Corrigés des activités

V.1. Activité 1

1. Relever dans ce programme les affectations et les expressions booléennes.

```
majeur = True #affectation
age = 18 # affectation
age == 18 #exp booléenne
```

```
## True
```

```
age < 18 #exp booléenne
```

```
## False
```

```
age != 12 #exp booléenne
```

```
## True
```

```
(majeur == True) or (age == 18) #exp booléenne
```

```
## True
```

```
age = 16 #affectation
(majeur == True) and (age >= 18) #exp booléenne
```

```
## False
```

```
not majeur #exp booléenne
```

```
## False
```

```
age != 18 #exp booléenne
```

```
## True
```

2. Justifier la valeur de chaque expression booléenne.

— `(majeur == True) or (age == 18)` on décompose majeur == True est Vrai car majeur est vrai. idem pour age qui vaut effectivement 18. L'expression revient à True OR True. Au moins une des deux propositions est vraies dont l'expression est vrai.
— `(majeur == True) and (age >= 18)` majeur est vrai donc majeur == True est vraie. age vaut 16 (ligne précédente) donc il n'est pas supérieur ou égal à 18. L'expression se ramène à vrai AND faux ce qui a pour valeur faux car pour que l'expression soit vraie il faut les deux vrai AND vrai.
— `not majeur` renvoie la négation de majeur. La négation de True, c'est False.
— `age != 18` est vrai si la valeur de age n'est pas 18. Ce qui ici est vrai car age vaut 16.

3. En modifiant les valeurs de `a` et `b` écrire la table de vérité de l'opérateur booléen `or`

```
a = True
b = False
a or b
```

```
## True
```

3. Complète le programme suivant afin que les expressions booléennes soient vraies :

```
a = 3
b = False
c = 5
(a <  c) or b
```

```
## True
```

```
not ( a != 3)
```

```
## True
```

```
(not b) and ( a == 3)
```

```
## True
```

4. L'opérateur booléen `or` est un ou inclusif. Dans l'expression `a or b` si a et b sont vraies alors l'expression est aussi vraie. On veut écrire une expression booléenne qui se comporte comme un ou exclusif c'est à dire en valant faux si a et b sont vraies. Par exemple : dans l'expression "fromage ou déssert ?" on ne valide pas le choix fromage ET déssert. Compléter le code ci-dessous.

```
fromage = False
dessert = False
# Expression booléeenne à écrire
(fromage or dessert) and not(fromage and dessert)
```

```
## False
```

5. A l'aide de votre programme écrire la table de vérité de cette opérateur OU-Exclusif

V.2. Activité 2

1. /
2.

```
expression_1 = True

if expression_1 : #si expression_1 est vrai alors
  print("Bloc 1") #affiche Bloc 1
  expression_1 = False #expression_1 prend la valeur Faux
```

```
## Bloc 1
```

```
print("Bloc 2") #affiche Bloc 2
```

```
## Bloc 2
```

```
if expression_1 : #Si application est vrai.. mais ici expression_1 vaut False donc on saute le bloc
  print("Bloc 3") #Pas exécuter car expression_1 est False

print("FIN") #Affiche Fin
```

```
## FIN
```

3.

```
age = 15
if age >= 18 :
  print("Accès autorisé")
```

4.

```
admin = True
age = 21
if (admin == True) and (age > 18):
  print("Connecté en tant qu' ADMIN !")
```

```
## Connecté en tant qu' ADMIN !
```

5.

```
heure = 2
prix = 10 + 5 * heure
if prix > 53:
   prix = 53
print(prix)
```

```
## 20
```

V.3. Activité 3

1.

```
note = 11
if note >=10:  #Si note >= 10 Alors Affiche "bonne note"
  print('Bonne note')
else:  #Sinon Alors Affiche "Mauvaise Note"
  print('Mauvaise note')
```

```
## Bonne note
```

2. Modifier le programme pour qu'il affiche "Mauvaise note".

```
note = 9
if note >=10:  #Si note >= 10 Alors Affiche "bonne note"
  print('Bonne note')
else:  #Sinon Alors Affiche "Mauvaise Note"
  print('Mauvaise note')
```

```
## Mauvaise note
```

3. Corriger l'erreur dans le programme suivant :

```
a = 12
if a >= 18:
  print('majeur')
else : # Pas de condition après else
  print('mineur')
```

```
## mineur
```

4. Compléter le programme suivant attribue une mention au bac.

— Inférieur à 10 : Recalé
— Entre 10 et 12 : Passable
— Entre 12 et 14 : Assez-bien
— Entre 14 et 16 : Bien
— 16 et plus : Très bien

```
note = 17
if note >= 16:
  print('Très bien')
elif note >= 14:
  print('Bien')
elif note >= 12 :
  print('Assez-bien')
elif note >= 10:
  print('Passable')
```

```
else:
  print('Recalé')
```

5. Modifier le programme précédent en inversant l'ordre des deux premiers tests conditionnels. Que se passe-t-il ? Justifie.

```
note = 17
if note >= 14:
  print('Bien')
elif note >= 16:
  print('Très bien')
```

```
## Bien
```

La première condition étant vraie la seconde n'est pas exécutée. C'est pourquoi avec note = 15, le programme affiche "Bien" et non pas "Très Bien" alors que note >=16 est vraie. 6. Ecrire un programme qui affiche "pair" si la valeur de la variable `n` est paire et "impaire" dans les autres cas. Pour savoir si un nombre est pair, on utilise le fait que le reste de la division par 2 de ce nombre vaut 0. On utilisera donc l'opérateur `%`.

```
n = 13
if n % 2 == 0:
  print(f'{n} est un entier pair')
else:
  print(f'{n} est un entier impair')
```

```
## 13 est un entier impair
```

V.4. Activité 4

1. Prédire et expliquer la valeur de `b` après exécution du programme suivant :

```
a=2
b=0
if a<0: #condition fausse
    b=1
elif a>0 and a<5: #condition vraie
    b=2 # b prend la valeur 2
else:
    b=3
```

2. Commenter l'exécution du programme suivant :

```
x = -3
if x < 0: #condition vraie
  if x % 2 == 0: #sous condition fausse
    print('pair et négatif')
  else:  #on exécute le bloc qui suit le else
    print('impair et négatif')
else: #pas exécuté car if correspondant vrai
    print("pair et c'est tout ce qui m'intéresse")
```

```
## impair et négatif
```

3. Modifier le programme afin qu'il affiche pair et positif ou impair et positif si c'est le cas.

```
x = 7
if x < 0:
    if x % 2 == 0:
```

```
        print('pair et négatif')
    else:
        print('impair et négatif')
else:
        if x % 2 == 0:
            print('pair et positif')
        else:
            print('impair et positif')
```

```
## impair et positif
```

4. Le programme suivant est buggé. On souhaite que lorsque l'utilisateur a plus de 18 ans et possède son code et le permis le programme affiche "vous pouvez conduire". Modifiez légèrement le programme afin de fixer cette erreur.

```
age = 19
code = True
permis = True

if age >= 14:
    print('vous pouvez apprendre à conduire un scooter')
elif code == True and age >=18:
    if permis == True:
        print('vous pouvez conduire une voiture')
    else:
        print('vous pouvez apprendre à conduire une voiture')
```

```
## vous pouvez apprendre à conduire un scooter
```

4. Ecrire un programme qui affiche la phrase "il va faire très froid / froid / frais / bon / un peu chaud / chaud / très chaud." selon la valeur de la variable `t` représentant la température maximale prévue. Vous choisirez les seuils entre chaque niveau de température.

```
t = 20
if t>30:
    mot = "très chaud"
elif t>25:
    mot = "chaud"
elif t>23:
    mot = "un peu chaud"
elif t>=20:
    mot = "bon"
elif t>17:
    mot = "frais"
elif t>10:
    mot = "froid"
else:
    mot = "très froid"
print(f"il va faire {mot} ")
```

```
## il va faire bon
```

5. Une année est bissextile si elle est divisible par 4. Cependant les siècles ne sont pas bissextiles, sauf les multiples de 400. Ecrire un programme qui affiche si une année est bisextile.

```
a = 2012
if a % 4 == 0 :
```

```
    if ((a % 100) == 0) and not(a%400 ==0):
        print('année pas bissextile')
    else:
        print('année bissextile')
else:
    print('année pas bissextile')
```

```
## année bissextile
```

Chapitre 3

Fonctions

I. Activité 1 : Généralités

1. Saisir le code suivant et l'analyser ligne à ligne

```
# Déclaration de la fonction bonjour_v1
def bonjour_v1():
    print('Bonjour')
# Appel de la fonction bonjour_v1
bonjour_v1()
```

II. Activité 2 : Plusieurs arguments

III. Activité 3 : Astuces

IV. Activité 4 : Modularité

Chapitre 4

Structures Itératives

Chapitre 5

Listes et tuples

Chapitre 6

QCM

I. Listes et tuples

Q1 - On veut affecter à t la valeur [[0,1,2], [3,4,5], [6,7,8], [9,10,11], [12,13,14]]. Pour cela on utilise le code suivant. Par quoi doit-on remplacer les pointillés …… ?

n = 5 p = 3 t = [[…… for j in range(p)] for i in range(n)]

Réponses :

A- i*j + j

B- p*i + j

C- p*j + i

D- i*(j+1)

Q2 - On considère le script suivant :

t = [2, 8, 9, 2] t[2] = t[2] + 5

Quelle est la valeur de t à la fin de son exécution ?

Réponses :

A- [2, 13, 9, 2]

B- [2, 8, 14, 2]

C- [7, 13, 14, 7]

D- [7, 13, 9, 2]

Q3 - On dispose dans le tableau annee2019 les températures mensuelles moyennes d'une région française. On exécute le script suivant :

annee2019 = [('janvier',6), ('février',6), ('mars',12), ('avril',20), ('mai',23), ('juin',25), ('juillet',29), ('août',25), ('septembre',22), ('octobre',15), ('novembre',11), ('décembre',7)]

m = annee2019[0][1] for mois in annee2019 : if (m > mois[1]) : m = mois[1]

Que contient la variable m à la fin de cette exécution ?

Réponses :

A- le mois le plus froid

B- le mois le plus chaud

C- la température moyenne la plus basse

D- la température moyenne la plus haute

Q4 - Quelle instruction permet d'affecter la liste [0,1,4,9,16] à la variable tableau ?

Réponses :

A- tableau = [i**2 for i in range(4)]

B- tableau = [i**2 for i in range(5)]

C- tableau = [i**2 for i in range(16)]

D- tableau = [i**2 for i in range(17)]

Q5 - Quelle est la valeur de la variable r à la fin de l'exécution du script suivant ?

t = (10,6,1,12,15) r = t[3] - t[1]

Réponses :

A- -9

B- 2

C- 3

D- 6

Q6 - On définit L = [4,25,10,9,7,13]. Quelle est la valeur de L[2] ?

Réponses :

A- 4

B- 25

C- 10

D- 9

Q7 - Quel est le type de l'expression f(4) si la fonction f est définie par :

def f(x) : return (x, x**2)

Réponses :

A- un entier

B- un flottant

C- une liste

D- un tuple

Q8 - On considère la liste de listes suivante :

tictactoe = [['X', 'O', 'O'], ['O', 'O', 'O'], ['O', 'O', 'X']]

Quelle instruction permet d'obtenir une diagonale de 'X' ?

Réponses :

A- tictactoe[3] = 'X'

B- tictactoe[4] = 'X'

C- tictactoe[1][1] = 'X'

D- tictactoe[2][2] = 'X'

Q9 - On définit ainsi une liste M :

M = [['A','B','C','D'], ['E','F','G','H'], ['I','J','K','L']]

Quelle expression vaut la chaîne de caractères 'H' ?

Réponses :

A- M[1][3]

B- M[3][1]

C- M(7)

D- M(8)

Q10 - On dispose d'une liste définie par L = [15,17,12,23]. Quelle est la valeur de L après l'instruction L[2] = 25 ?

Réponses :

A- [15,25,12,23]

B- [15,17,25,12,23]

C- [15,17,25,23]

D- [15,17,12,25,23]

Q11 - Soient n et p deux entiers au moins égaux à 2. On définit une liste de listes t par le code suivant :

(n et p sont initialisés dans les lignes précédentes)

t = [[0 for j in range(p)] for i in range(n)]

for k in range(n*p) : t[k%n][k%p] = k

Une et une seule des affirmations suivantes est fausse. Laquelle ?

Réponses :

A- La liste t contient des entiers tels que 0 <= k < n x p

B- Pour tout j tel que 0 <= j < n-1, t[j][0] est un multiple de p.

C- La liste t[0] contient des entiers qui sont tous multiples de n.

D- Pour tout j tel que 0 <= j < n-1, t[0][j] est un multiple de p.

Q12 - Que vaut l'expression [2*k for k in range(5)] ?

Réponses :

A- [0,2,4,6,8]

B- [2,4,6,8,10]

C- [1,2,4,8,16]

D- [2,4,8,16,32]

Q13 - On exécute le code suivant :

A = [[1,2,3], [4,5,6], [7,8,9]] B = [[0,0,0], [0,0,0], [0,0,0]] for i in range(3) : for j in range(3) : B[i][j] = A[j][i]

Que vaut B à la fin de l'exécution ?

Réponses :

A- rien du tout, le programme déclenche une erreur d'exécution

B- [[3,2,1], [6,5,4], [9,8,7]]

C- [[1,4,7], [2,5,8], [3,6,9]]

D- [[7,8,9], [4,5,6], [1,2,3]]

Q14 - Après l'affectation suivante :

alphabet = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']

quelle est l'expression qui permet d'accéder à la lettre E ?

Réponses :

A- alphabet.E

B- alphabet['E']

C- alphabet[4]

D- alphabet[5]

Q15 - On définit : L = [10,9,8,7,6,5,4,3,2,1]. Quelle est la valeur de L[L[3]] ?

Réponses :

A- 3

B- 4

C- 7

D- 8

Q16 - On considère le code suivant :

t = [0, 3, 5, 7, 9] t[9] = 3 + t[5]

Que vaut t à la fin de son exécution ?

Réponses :

A- [0, 3, 5, 7, 9]

B- [0, 3, 5, 7, 9, 3]

C- [0, 3, 5, 7, 9, 8]

D- l'exécution déclenche une erreur

Q17 - On définit :

resultat = [i*2 for i in range(10)]

Quelle est la valeur de resultat ?

Réponses :

A- [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

B- [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]

C- [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

D- [2, 4, 6, 8, 10, 12, 14, 16, 18]

Q18 - On considère la fonction suivante :

def somme(tab) : s = 0 for i in range(len(tab)) : …… return s

Par quelle instruction faut-il remplacer les points de suspension pour que l'appel somme([10,11,12,13,14]) renvoie 60 ?

Réponses :

A- s = tab[i]

B- s = s + tab[i]

C- tab[i] = tab[i] + s

D- s = s + i

Q19 - On exécute le script suivant.

m = [] for i in range(5) : n = [] for j in range(3) : n.append(i*j) m.append(n)

Quelle est la valeur de m à la fin de son exécution ?

Réponses :

A- [[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]

B- [[0, 0, 0], [0, 1, 2], [0, 2, 4], [0, 3, 6], [0, 4, 8]]

C- [[1, 1, 1], [2, 4, 6], [3, 6, 9], [4, 8, 12], [5, 10, 15]]

D- [[1, 1, 1, 1, 1], [2, 4, 6, 8, 10], [3, 6, 9, 12, 15], [4, 8, 12, 16, 20], [5, 10, 15, 20, 25]]

Q20 - On considère le code suivant :

def feed(t) : for i in range(len(t)) : t[i] = 0 return t

Que renvoie feed([12, 24, 32]) ?

Réponses :

A- [120, 240, 320]

B- [0, 0, 0]

C- [ ]

D- [0]

Q21 - L est une liste d'entiers. On définit la fonction suivante :

def f(L) : m = L[0] for x in L : if x > m : m = x return m

Que calcule cette fonction ?

Réponses :

A- le maximum de la liste L passée en argument

B- le minimum de la liste L passée en argument

C- le premier terme de la liste L passée en argument

D- le dernier terme de la liste L passée en argument

Q22 - On considère le code suivant :

t = [1, 6, 8, 3, 21] u = [x for x in t if x > 3]

Que vaut u à la fin de son exécution ?

Réponses :

A- [1, 6, 8, 21]

B- [6, 8, 3, 21]

C- [6, 8, 21]

D- [1, 3, 6, 21]

Q23 - On exécute le code suivant :

t = [1,2,3,4,5,6,7,8,9] v = [c for c in t if c%3 == 0]

Quelle est la valeur de la variable v à la fin de cette exécution ?

Réponses :

A- 18

B- [1,4,7]

C- [3,6,9]

D- [1,2,3,4,5,6,7,8,9]

Q24 - On a défini : T = [[1,2,3], [4,5,6], [7,8,9]].

Quelle expression parmi les suivantes a pour valeur le nombre 8 ?

Réponses :

A- T[1,2]

B- T[1][2]

C- T[2,1]

D- T[2][1]

Q25 - On définit : L = [[“lundi”,10,0.87], [“mardi”,11,0.82], [“mercredi”,12,0.91]]

Quel est le type de la variable a définie par a = L[1][2] ?

Réponses :

A- nombre entier

B- liste

C- nombre flottant

D- chaîne de caractères

Q26 - n définit la liste L ainsi : L = [[1], [1,2], [1,2,3]]

Des égalités suivantes, une seule est fausse. Laquelle ?

Réponses :

A- len(L[0]) == 1

B- len(L) == 6

C- len(L[2]) == 3

D- L[2][2] == 3

Q27 - On définit L = [[1,2,3,4,5],[6,7,8,9,10],[11,12,13,14,15]].

Quelle est la valeur de L[0][2] ?

Réponses :

A- 2

B- 3

C- 11

D- 12

Q28 - On considère deux entiers strictement positifs L et C. On note n = L*C leur produit et on écrit la fonction suivante, qui construit un tableau de L lignes et C colonnes, contenant les entiers consécutifs de 0 à n-1 :

def construitTable(L,C) : t = [] for i in range(L) : ligne = [] for j in range(C) : …… t.append(ligne) return t

Par exemple, l’appel construitTable(2,3) doit renvoyer la table : [[0, 1, 2],[3, 4, 5]]

Que faut-il écrire à la place des points de suspension pour obtenir ce résultat ?

Réponses :

A- ligne.append(i + C*j)

B- ligne.append(L*i + j)

C- ligne.append(i + L*j)

D- ligne.append(C*i + j)

Q29 - La fonction ci-dessous prend en argument deux nombres entiers.

def f(n1,n2) : etendue = max(n1,n2)-min(n1,n2) moyenne = (n1+n2)/2 return etendue,moyenne

Quel est le type de la valeur renvoyée par un appel à cette fonction ?

Réponses :

A- un entier

B- un réel (ou flottant)

C- un tuple

D- une liste

Q30 - Si on tape dans la console d'exécution la commande :

[1,4,3] + [2,4,5]

qu'obtient-on ?

Réponses :

A- [3, 8, 8]

B- [19]

C- [1, 4, 3, 2, 4, 5]

D- un message d'erreur car l'addition n'est pas compatible avec les listes

Q31 - Quelle est la valeur de l'expression [(a,b) for a in range(3) for b in range(3) if a > b] ?

Réponses :

A- [(a,b),(a,b),(a,b),(a,b),(a,b),(a,b),(a,b),(a,b),(a,b)]

B- [(0,0),(0,1),(0,2),(1,0),(1,1),(1,2),(2,0),(2,1),(2,2)]

C- [(1,0),(2,0),(2,1)]

D- [(0,0),(0,1),(0,2),(1,0),(1,1),(1,2),(1,0),(1,1),(1,2)]

Q32 - Quelle est la valeur de l'expression [[i for i in range(5)] for j in range(3)] ?

Réponses :

A- [[0, 1, 2], [0, 1, 2], [0, 1, 2], [0, 1, 2], [0, 1, 2]]

B- [[0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4]]

C- [[0, 1, 2, 3], [0, 1, 2, 3], [0, 1, 2, 3], [0, 1, 2, 3], [0, 1, 2, 3]]

D- [[0, 1, 2, 3, 4, 5], [0, 1, 2, 3, 4, 5], [0, 1, 2, 3, 4, 5]]

Q33 - On définit une grille G remplie de 0, sous la forme d'une liste de listes, où toutes les sous-listes ont le même nombre d'éléments.

G =[[0, 0, 0, ..., 0], [0, 0, 0, ..., 0], [0, 0, 0, ..., 0] .... [0, 0, 0, ..., 0]]

On appelle hauteur de la grille le nombre de sous-listes contenues dans G et largeur de la grille le nombre d'éléments dans chacune de ces sous-listes. Comment peut-on les obtenir ?

Réponses :

A- hauteur = len(G[0])

largeur = len(G)

B- hauteur = len(G)

largeur = len(G[0])

C- hauteur = len(G[0])

largeur = len(G[1])

D- hauteur = len(G[1])

largeur = len(G[0])

Q34 - Laquelle des expressions suivantes a-t-elle pour valeur la liste des carrés des premiers entiers qui ne sont pas multiples de 5 ?

Réponses :

A- [x*x for x in range (11) if x//5 != 0]

B- [x*x if x%5 != 0 for x in range (11)]

C- [x*x if x//5 != 0 for x in range (11)]

D- [x*x for x in range (11) if x%5 != 0]

Q35 - On définit :

tab = [(‘Léa’, 14), (‘Guillaume’, 12), (‘Anthony’, 16), (‘Anne’, 15)]

Quelle est la valeur de l’expression [x[0] for x in tab if x[1]>=15] ?

Réponses :

A- [(‘Anthony’, 16), (‘Anne’, 15)]

B- [‘Anthony’, ‘Anne’]

C- [16, 15]

D- TypeError : ‘tuple’ object is not callable

Q36 - On définit : matrice = [[1,2,3], [4,5,6], [7,8,9], [10,11,12]].

Quelle est la valeur de matrice[1][2] ?

Réponses :

A- 2

B- 4

C- 6

D- 8

Q37 - On crée la liste suivante :

t = [[1,2,3,4], [5,6,7,8], [9,10,11,12]]

Que vaut t[1][2] :

Réponses :

A- 2

B- 7

C- 10

D- on obtient un message d’erreur “indexError : list index out of range”

Q38 - Le premier élément d’une liste Python L est noté :

Réponses :

A- L(0)

B- L(1)

C- L[0]

D- L[1]

Q39 - Quelle expression Python a pour valeur la liste [1,3,5,7,9,11] ?

Réponses :

A- [2*i - 1 for i in range(6)]

B- [2*i + 1 for i in range(6)]

C- [2*i + 1 for i in range(5)]

D- [2*i - 1 for i in range(7)]

Q40 - Après avoir défini m = [[1, 2, 3], [4, 5, 6]]

laquelle des quatre expressions suivantes a la valeur 4 ?

Réponses :

A- m[0][1]

B- m[1][0]

C- m(0,1)

D- m(1,0)

Q41 - On définit tableau = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]. Quelle est la valeur de tableau[2][1] ?

Réponses :

A- 2

B- 4

C- 6

D- 8

Q42 - On s'intéresse à la valeur 14 présente dans la liste suivante :

T = [[1,2,3,4,5], [6,7,8,9,10], [11,12,13,14,15], [16,17,18,19,20]].

Quelle expression vaut 14 parmi les suivantes ?

Réponses :

A- T[2][3]

B- T[3][4]

C- T[3][2]

D- T[4][3]

Q43 - On définit : t = [2, 8, 9, 2]

Quelle est la valeur de l'expression [x*x for x in t] ?

Réponses :

A- une erreur

B- [[2, 8, 9, 2], [2, 8, 9, 2]]

C- [2, 8, 8, 9, 9, 9, 2, 2, 2, 2]

D- [4, 64, 81, 4]

Q44 - De quelle expression la liste suivante est-elle la valeur ?

[[0,0,0,0], [1,1,1,1], [2,2,2,2]]

Réponses :

A- [[i] * 4 for i in range(4)]

B- [[i] * 3 for i in range(4)]

C- [[i] * 4 for i in range(3)]

D- [[i] * 3 for i in range(3)]

Q45 - On définit :

notes = [('Toto', 20), ('John', 12), ('Johnny', 2), ('Superman', 16)]

Quelle est l'expression donnant la note de Superman ?

Réponses :

A- notes[4][2]

B- notes[3][1]

C- notes[Superman]

D- notes['Superman']

Q46 - Quelle est la valeur de l'expression [[n,n+2] for n in range(3)] ?

Réponses :

A- [0,2,1,3,2,4]

B- [1,3,2,4,3,5]

C- [[0,2],[1,3],[2,4]]

D- [[1,3],[2,4],[3,5]]

Q47 - On construit une matrice par compréhension :

M = [[i*j for j in range(4)] for i in range(4)]

Laquelle des conditions suivantes est-elle vérifiée ?

Réponses :

A- M[4][4] == 16

B- M[0][1] == 1

C- M[2][3] == 6

D- M[1][2] == 3

Q48 - Quelle est l'expression qui a pour valeur la liste [1,4,9,16,25,36] ?

Réponses :

A- { n*n for n in range(1,7) }

B- { n*n for n in range(6) }

C- [n*n for n in range(1,7)]

D- [n*n for n in range(6)]

Q49 - On dispose d'une table tab constituée d'une liste de trois sous-listes contenant chacune quatre caractères.

tab=[['A', 'B', 'C', 'D'], ['E', 'F', 'G', 'H'], ['I', 'J', 'K', 'L']]

Parmi les propositions suivantes, laquelle permet de convertir cette table en une liste L contenant dans l'ordre, ligne par ligne, les 12 caractères de tab ?

à la fin, on a l'égalité : L == ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L']

Réponses :

A-

L = [] for i in range(3) : for j in range(4) : L.append(tab[i][j])

B-

L = [] for i in range(4) : for j in range(3) : L.append(tab[i][j])

C-

L = [] for i in range(3) : L.append(tab[i])

D-

L = [] for i in range(4) : L.append(tab[i])

Q50 - On dispose d'une liste définie par L = [[1,2,3],[4,5,6],[7,8,9]].

Quelle est la valeur de L[1][2] ?

Réponses :

A- 2

B- 4

C- 6

D- 8

Q51 - Quelle est la valeur de l'expression [[i,2*i] for i in range(3)] ?

Réponses :

A- [0,0,1,2,2,4]

B- [[0,0],[1,2],[2,4]]

C- [1,2,2,4,3,6]

D- [[1,2],[2,4],[3,6]]

Q52 - Voici une définition incomplète d'une fonction qui renvoie le couple du quotient et du reste de la division euclidienne :

def divEuclid(n,d) : ''renvoie le couple formé du quotient et du reste dans la division de n par d" ' q = 0 while n-d > 0 : q = q + 1 n = n - d …….

Par quelle instruction faut-il remplacer la ligne en pointillés pour que l'appel

(quotient,reste) = divEuclid(15,6)

affecte les valeurs attendues ?

Réponses :

A- (q,n)

B- (quotient,reste)

C- return (q,n)

D- return (quotient,reste)

Q53 - On considère le tableau suivant : L = [[1,2,3],[4,5,6],[7,8,9]].

Quelle est la valeur de L[1][0] ?

Réponses :

A- 1

B- 2

C- 4

D- 7

Q54 - Quelle est la valeur de l'expression [(i,i+1) for i in range(2)] ?

Réponses :

A- [0,1,1,2]

B- [(1,2),(2,3)]

C- [(0,1),(1,2)]

D- [[0,1],[1,2]]

Q55 - Soit le code ci-dessous :

tableau = [5,8,6,9] a = tableau[2]

Après son exécution, quelle valeur contient la variable a ?

Réponses :

A- 2

B- 6

C- 8

D- [5, 8]

Q56 - On définit la variable suivante : citation = "Les nombres gouvernent le monde". Quelle est la valeur de l'expression citation[5 :10] ?

Réponses :

A- "ombre"

B- "ombres"

C- "nombre"

D- "nombres"

Q57 - Quelle expression a pour valeur la liste [7,14,21,28,35,42,49,56,63] ?

Réponses :

A- [7*k for k in range(9)]

B- [7*k for k in range(1,10)]

C- [7*k for k in range(10)]

D- [[7*k for k in range(1,9)]

Q58 - Quelle est la valeur de l'expression [[0] * 3 for i in range(2)] ?

Réponses :

A- [[0,0], [0,0], [0,0]]

B- [[0,0,0], [0,0,0]]

C- [[0.000], [0.000]]

D- [[0.00], [0.00], [0.00]]

Q59 - Quelle est la valeur de la variable S à la fin de l'exécution du script suivant ?

res = [[1,2,3], [4,5,6], [7,8,9]] S = 0 for i in range(3) : S = S + res[i][2]

Réponses :

A- 12

B- 15

C- 18

D- 24

Q60 - On représente un plateau de jeu d'échec par une liste de listes dans laquelle on place des 1 pour représenter une case où se trouve une tour et des 0 pour représenter les cases vides.

Par exemple le code

echiquier = [[0 for i in range(8)] for j in range(8)] echiquier[2][0] = 1 echiquier[3][1] = 1

représente la situation de la figure ci-dessous.

Deux tours sont en prise si elles se trouvent sur une même ligne ou sur une même colonne.

Parmi les codes suivants, lequel permet de vérifier que la tour placée en ligne i et en colonne j n'est en prise avec aucune tour placée dans les colonnes à sa gauche ?

Réponses :

A-

def ok(echiquier,i,j) : for col in range(i) : if echiquier[i][col] == 1 : return False return True

B-

def ok(echiquier,i,j) : for lig in range(i) : if echiquier[lig][j] == 1 : return False return True

C-

def ok(echiquier,i,j) : for col in range(j) : if echiquier[i][col] == 1 : return False return True

D-

def ok(echiquier,i,j) : for lig in range(j) : if echiquier[lig][j] == 1 : return False return True

Q61 - On dispose d'une liste L :

L = [6, 2, 8, 24, 3, 6, 7, 8]

Quelle est la valeur de M après exécution du code suivant ?

p = 8 M = [x for x in L if x<p] + [x for x in L if x==p] + [x for x in L if x>p]

Réponses :

A- [2,3,6,6,7,8,8,24]

B- [6,2,3,6,7,8,8,24]

C- [6,2,8,24,3,6,7,8]

D- [[6,2,3,6,7],[8,8],[24]]

Q62 - On définit en Python la fonction suivante :

def f(L) : U = [] for i in L : U.append(i**2 - 1) return U

Que vaut f([-1, 0, 1, 2]) ?

Réponses :

A- [0, 0, 1, 3]

B- [-1, 0, 0, 3]

C- [0, -1, 0, 3]

D- [-3, -1, 1, 3]

Q63 - On définit L = [2,3,5,7,-4].

En demandant la valeur de L[5], qu'obtient-on ?

Réponses :

A- -4

B- 2

C- 3

D- une erreur

Q64 - Considérons le tableau suivant :

tableau = [[1,2],[3,4],[5,6]]

Quelle est la valeur de l'expression tableau[2][1] ?

Réponses :

A- 3

B- 6

C- [3,4],[1,2]

D- [5,6],[2,4]

Q65 - Laquelle des quatre expressions suivantes a-t-elle pour valeur la liste [1, 2, 5, 10] ?

Réponses :

A- [i for i in range(4) if i<2]

B- [i for i in range(4)]

C- [i*i + 1 for i in range(4)]

D- [i*i* - 2i + 2 for i in range(4)]

Q66 - Considérons le tableau suivant :

tableau = [[i+2*j for j in range(4)] for i in range(4)]

Quelle est la valeur de l'expression tableau[1] ?

Réponses :

A- [0,1,2,3]

B- [1,2,3,4]

C- [0,2,4,6]

D- [1,3,5,7]

Q67 - On définit ainsi le tableau t = [[1,5,7], [8,4,2], [3,9,6]]

Quel jeu d'indices permet d'obtenir l'élément "9" de ce tableau ?

Réponses :

A- t[3][2]

B- t[2][3]

C- t[1][2]

D- t[2][1]

Q68 - On définit une liste : L = [1, 1, 2, 9, 3, 4, 5, 6, 7].

Quelle expression a-t-elle pour valeur la liste [4, 16, 36] ?

Réponses :

A- [(x * x) % 2 == 0 for x in liste]

B- [x for x in liste if x % 2 == 0]

C- [x * x for x in liste]

D- [x * x for x in liste if x % 2 == 0]

Q69 - Parmi les propositions suivantes, laquelle permet de créer en Python la liste des nombres impairs de 1 à 399 (inclus) ?

Réponses :

A-

impairs = [1 + nb*2 for nb in range(200)]

B-

for nb in range(400) : impairs = 1 + 2 * nb

C-

impairs = [i + 2 for i in range(1,200)]

D-

impairs = [1, 3, 5, 7, 9] * 40

Q70 - On considère le code suivant :

def f(L) : return [x*x for x in L if x%2 == 1]

carre = f([0,1,2,3,4,5,6,7,8,9])

Que vaut carre à la fin de son exécution ?

Réponses :

A- [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

B- [0, 4, 16, 36, 64]

C- [1, 9, 25, 49, 81]

D- [0, 2, 4, 6, 8, 10, 12, 14, 16, 19]

Q71 - L'opérateur % calcule le reste de la division euclidienne de l'opérande de gauche par l'opérande de droite. Par exemple : 7 % 3 vaut 1, 15 % 5 vaut 0 et 18 % 4 vaut 2.

On crée la liste suivante :

t = [x for x in range(2,12) if x % 2 == 1]

Que vaut t :

Réponses :

A- [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

B- [1, 2, 3, 4, 5]

C- [3, 5, 7, 9, 11]

D- [0, 1, 0, 1, 0, 1, 0, 1, 0, 1]

Q72 - Quelle expression permet d'accéder à la valeur 'hello' après qu'on a défini

L = [[‘a’,‘b’,‘c’],[‘bonjour’,‘hello’]]

Réponses :

A- L[5]

B- L[1][1]

C- L[2][2]

D- L[‘hello’]

Q73 - t1 est un tableau à n lignes et n colonnes. On souhaite remplir un tableau t2 de mêmes dimensions que t1 avec les contraintes suivantes : les lignes de t2 sont les colonnes de t1 et les colonnes de t2 sont les lignes de t1.

Par quelle instruction faut-il remplacer la ligne en pointillées du code suivant ?

for i in range(n) : for j in range(n) : ……

Réponses :

A- t1[i][j] = t2[j][i]

B- t2[j][i] = t1[j][i]

C- t1[j][i] = t2[i][j]

D- t2[i][j] = t1[j][i]

Q74 - Soit une liste définie de la manière suivante : liste = [18, 23, 45, 38, 12]

On exécute l’instruction liste.append(45), la liste a alors pour valeur :

Réponses :

A- [18, 23, 38, 12, 45]

B- [18, 23, 38, 12]

C- [45, 18, 23, 45, 38, 12]

D- [18, 23, 45, 38, 12, 45]

Q75 - On dispose d’une liste L constituée de 12 caractères.

L = [‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, ‘G’, ‘H’, ‘I’, ‘J’, ‘K’, ‘L’]

Parmi les propositions suivantes, laquelle permet de convertir cette liste en une table tab constituée d’une liste de trois sous-listes contenant chacune quatre caractères contenant dans l’ordre, et contenant les 12 caractères de L dans l’ordre.

à la fin, on a l’égalité : tab == [[‘A’, ‘B’, ‘C’, ‘D’], [‘E’, ‘F’, ‘G’, ‘H’], [‘I’, ‘J’, ‘K’, ‘L’]]

Réponses :

A-

tab = [] for i in range(4) : temp = [] for j in range(3) : temp.append(L[4*i + j]) tab.append(temp)

B-

tab = [] for i in range(4) : temp = [] for j in range(3) : temp.append(L[3*i + j]) tab.append(temp)

C-

tab = [] for i in range(3) : temp = [] for j in range(4) : temp.append(L[3*i + j]) tab.append(temp)

D-

tab = [] for i in range(3) : temp = [] for j in range(4) : temp.append(L[4*i + j]) tab.append(temp)

Q76 - On exécute l’instruction suivante :

T = [[12,13,14,15], [24,25,26,27], [35,36,49,33], [61,53,55,58]]

Quelle expression parmi les quatre suivantes a pour valeur 26 ?

Réponses :

A- T[1][2]

B- T[2][1]

C- T[2][3]

D- T[3][2]

Q77 - Parmi les scripts suivants, un seul ne permet pas de générer le tableau [0,2,4,6,8,10,12,14,16,18] noté T.

Quel est ce script fautif ?

Réponses :

A-

T = [] for k in range(10) : T.append(2*k)

B-

T = [0] * 10 for k in range(9) : T[k+1] = 2*(k+1)

C-

T = [2*k for k in range(10)]

D-

T = [0] * 10 for k in range(0) : T[k+1] = 2*T[k]

Q78 - On dispose d'une liste L d'entiers rangés en ordre croissant.

On désire connaître le nombre de valeurs distinctes contenues dans cette liste.

Parmi les quatre fonctions proposées, laquelle ne donne pas le résultat attendu ?

Réponses :

A-

def nombreDistincts(L) : n = 1 for i in range(1,len(L)) : if L[i] != L[i-1] : n = n + 1 return n

B-

def nombreDistincts(L) : n = 1 for i in range(0,len(L)-1) : if L[i] != L[i+1] : n = n + 1 return n

C-

def nombreDistincts(L) : n = 0 for i in range(0,len(L)-1) : if L[i] != L[i+1] : n = n + 1 return n

D-

def nombreDistincts(L) : n = 0 for i in range(1,len(L)) : if L[i] != L[i-1] : n = n + 1 return n

Q79 - On définit ainsi une liste M :

M = [['A','B','C','D'], ['E','F','G','H'], ['I','J','K','L']]

Que vaut l'expression M[2][1] ?

Réponses :

A- 'G'

B- 'J'

C- 'E'

D- 'B'

Q80 - Quelle est la valeur de l'expression [2**i for i in range(5)] ?

Réponses :

A- [0,1,4,9,16]

B- [1,4,9,16,25]

C- [0,2,4,6,8]

D- [1,2,4,8,16]

Q81 - Quelle affectation permet de donner à L la valeur [1,9,25,49,81] ?

Réponses :

A- L = [i*2 for i in range(9) if i%2 == 0]

B- L = [i**2 for i in range(10) if i%2 == 0]

C- L = [i**2 for i in range(10) if i%2 == 1]

D- L = [i**2 for i in range(10) if i//2 == 1]

Q82 - On définit : T = [7*n for n in range(10)].

Quelle est la valeur de l'expression T[7] ?

Réponses :

A- 42

B- 49

C- 56

D- 70

Q83 - On définit en Python la fonction suivante :

def f(L) : S = [] for i in range(len(L)-1) : S.append(L[i] + L[i+1]) return S

Quelle est la liste renvoyée par f([1, 2, 3, 4, 5, 6]) ?

Réponses :

A- [3, 5, 7, 9, 11, 13]

B- [1, 3, 5, 7, 9, 11]

C- [3, 5, 7, 9, 11]

D- cet appel de fonction déclenche un message d'erreur

Q84 - On exécute le script suivant :

def quoi(liste) : maListe = [] for i in range(len(liste)) maListe.append(liste[i][0]) return maListe

L = [[5,8,12,1], [20,11,3,8], [3,12,1,4], [2,13,17,3]] m = quoi(L)

Que contient la variable m à la fin de cette exécution ?

Réponses :

A- 26

B- 30

C- [5, 20, 3, 2]

D- [5, 8, 12, 1]

Q85 - On souhaite construire une table de 4 lignes de 3 éléments que l'on va remplir de 0. Quelle syntaxe Python utilisera-t-on ?

Réponses :

A- [[0] * 3 for i in range (4)]

B- for i in range (4) [0] * 3

C- [0] * 3 for i in range (4)

D- [for i in range (4) [0] * 3]

Q86 - On considère la liste de p-uplets suivante :

Table = [('Grace','Hopper','F',1906),('Tim', 'Berners-Lee', 'H', 1955),('Ada', 'Lovelace', 'F', 1815), ('Alan', 'Turing', 'H', 1912)]

où chaque p-uplet représente un informaticien ou une informaticienne célèbre ; le premier élément est son prénom, le deuxième élément son nom, le troisième élément son sexe ('H' pour un homme, 'F' pour une femme) et le quatrième élément son année de naissance (un nombre entier entre 1000 et 2000).

On définit une fonction :

def fonctionMystere(table) : mystere = [] for ligne in table : if ligne[2] == 'F' : mystere.append(ligne[1]) return mystere

Que vaut fonctionMystere(table) ?

Réponses :

A- ['Grace', 'Ada']

B- [('Grace', 'Hopper', 'F', 1906), ('Ada', 'Lovelace', 'F', 1815)]

C- ['Hopper', 'Lovelace']

D- []

Q87 - Soit le tableau défini de la manière suivante : tableau = [[1,3,4],[2,7,8],[9,10,6],[12,11,5]]

On souhaite accéder à la valeur 12, on écrit pour cela :

Réponses :

A- tableau[4][1]

B- tableau[1][4]

C- tableau[3][0]

D- tableau[0][3]

Q88 - Quelle est la valeur de la variable table à la fin de l'exécution du script suivant :

table = [[1, 2, 3], [1, 2, 3], [1, 2, 3], [1, 2, 3]] table [1][2] = 5

Réponses :

A- [[1, 5, 3], [1, 2, 3], [1, 2, 3], [1, 2, 3]]

B- [[1, 2, 3], [5, 2, 3], [1, 2, 3], [1, 2, 3]]

C- [[1, 2, 3], [1, 2, 5], [1, 2, 3], [1, 2, 3]]

D- [[1, 2, 3], [1, 2, 3], [1, 2, 3], [1, 5, 3]]

Q89 - On exécute le code suivant :

a = [5, 4, 3, 4, 7] a.append(4)

Quelle est la valeur de la variable a à la fin de cette exécution ?

Réponses :

A- 2

B- [4, 4]

C- [5, 4, 3, 4, 7, 4]

D- True

Q90 - Quelle est la valeur de la variable image après exécution du programme Python suivant ?

image = [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]] for i in range(4) : for j in range(4) : if (i+j) == 3 : image[i][j] = 1

Quelle est la valeur de la variable a à la fin de cette exécution ?

Réponses :

A- [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [1, 1, 1, 1]]

B- [[0, 0, 0, 1], [0, 0, 0, 1], [0, 0, 0, 1], [0, 0, 0, 1]]

C- [[0, 0, 0, 1], [0, 0, 1, 0], [0, 1, 0, 0], [1, 0, 0, 0]]

D- [[0, 0, 0, 1], [0, 0, 1, 1], [0, 1, 1, 1], [1, 1, 1, 1]]

Q91 - Laquelle de ces listes de chaînes de caractères est triée en ordre croissant ?

Réponses :

A- [“112”, “19”, “27”, “45”, “8”]

B- [“8”, “19”, “27”, “45”, “112”]

C- [“8”, “112”, “19”, “27”, “45”]

D- [“19”, “112”, “27”, “45”, “8”]

Q92 - On a défini :

mendeleiev = [[‘H’,‘’, ‘’,‘’,‘’,‘’,‘’,‘’,‘He’], [‘Li’,‘Be’,‘B’,‘C’,‘N’,‘O’,‘Fl’,‘Ne’], [‘Na’,‘Mg’,‘Al’,‘Si’,‘P’,‘S’,‘Cl’,‘Ar’], ……]

Une erreur s’est glissée dans le tableau, car le symbole du Fluor est F et non Fl. Quelle instruction permet de rectifier ce tableau ?

Réponses :

A- mendeleiev.append(‘F’)

B- mendeleiev[1][6] = ‘F’

C- mendeleiev[6][1] = ‘F’

D- mendeleiev[-1][-1] = ‘F’

Q93 - On exécute le script suivant :

asso = [] L = [[‘marc’,‘marie’], [‘marie’,‘jean’], [‘paul’,‘marie’], [‘marie’,‘marie’], [‘marc’,‘anne’]] for c in L : if c[1]==’marie’ : asso.append(c[0])

Que vaut asso à la fin de l’exécution ?

Réponses :

A- [‘marc’, ‘jean’, ‘paul’]

B- [[‘marc’,‘marie’], [‘paul’,‘marie’], [‘marie’,‘marie’]]

C- [‘marc’, ‘paul’, ‘marie’]

D- [‘marie’, ‘anne’]

Q94 - On exécute le script suivant :

a = [1, 2, 3] b = [4, 5, 6] c = a + b

Que contient la variable c à la fin de cette exécution ?

Réponses :

A- [5,7,9]

B- [1,4,2,5,3,6]

C- [1,2,3,4,5,6]

D- [1,2,3,5,7,9]

Q95 - On a défini :

mendeleiev = [[‘H’,‘:’, ‘:’,‘:’,‘:’,‘:’,‘:’,‘He’], [‘Li’,‘Be’,‘B’,‘C’,‘N’,‘O’,‘Fl’,‘Ne’], [‘Na’,‘Mg’,‘Al’,‘Si’,‘P’,‘S’,‘Cl’,‘Ar’], ……]

Comment construire la liste des gaz rares, c’est-à-dire la liste des éléments de la dernière colonne ?

Réponses :

A- gaz_rares = [periode[7] for periode in mendeleiev]

B- gaz_rares = [periode for periode in mendeleiev[7]]

C- gaz_rares = [periode for periode[7] in mendeleiev]

D- gaz_rares = [periode[8] for periode in mendeleiev]

Q96 - On a défini deux tables de données :

data1 = [(‘Bruce’, ’Wayne’), (‘Chuck’, ‘Norris’), (‘Bruce’, ‘Lee’), (‘Clark’, ‘Kent’)] data2 = [(‘Diana’, ’Prince’), (‘Chuck’, ‘Norris’), (‘Peter’, ‘Parker’)]

Quelle instruction permet de construire une table data regroupant l’ensemble des informations de data1 et data2 ?

Réponses :

A- data = data1 + data2

B- data == data1 + data2

C- data = [element for element in data1 or data2]

D- data = [data1] + [data2]

Q97 - On déifnit la fonction suivante qui prend en argument un tableau non vide d’entiers.

def f(T) : s = 0 for k in T : if k == 8 : s = s+1 if s > 1 : return True else : return False

Dans quel cas cette fonction renvoie-t-elle la valeur True ?

Réponses :

A- dans le cas où 8 est présent au moins une fois dans le tableau T

B- dans le cas où 8 est présent au moins deux fois dans le tableau T

C- dans le cas où 8 est présent exactement une fois dans le tableau T

D- dans le cas où 8 est présent exactement deux fois dans le tableau T

Q98 - Quelle est la valeur de l’expression [2*k + 1 for k in range(4)] ?

Réponses :

A- [1,3,5,7]

B- [0,1,2,3]

C- [3,5,7,9]

D- [1,2,3,4]

Q99 - On exécute le code suivant :

collection = [(‘Renault’, ‘4L’, 1974, 30), (‘Peugeot’, ‘504’, 1970, 82), (‘Citroën’, ‘Traction’, 1950, 77)]

Que vaut collection[1][2] ?

Réponses :

A- 1970

B- ‘4L’

C- (‘Peugeot’, ‘504’, 1970, 82)

D- (‘Renault’, ‘4L’, 1974, 30)

Q100 - On exécute le code suivant :

def maxi(t) : m = t[0] for x in t : if x[1] >= m[1] : m = x return m

L = [(‘Alice’, 17), (‘Barnabé’, 17),(‘Casimir’, 17), (‘Doriane’, 17),(‘Emilien’, 14), (‘Fabienne’, 16)]

Quelle est alors la valeur de maxi(L) ?

Réponses :

A- (‘Alice’,17)

B- (‘Doriane’,17)

C- (‘Fabienne’,17)

D- (‘Emilien’,14)

Q101 - Dans une course de chevaux, chaque participant est représenté par un numéro de dossard unique , son nom et sa ville sous la forme d’un couple formé d’un entier et d’une liste : (dossard, [nom, ville]).

Les données de quelques participants sont réunies au sein de la liste course ci-dessous :

course = [(5,[‘Mistral’,‘Lille’]), (3,[‘Zéphir’,‘Paris’]),(7,[‘Ouragan’,‘Bordeaux’]), ……, …….]

Quelle expression permet d’obtenir la ville du cheval nommé Zéphir ?

Réponses :

A- course[1][1][1]

B- course[1][1][2]

C- course[1][2][1]

D- course[2][1][1]

Q102 - Quelle est la valeur de la variable t1 à la fin de l’exécution du script suivant :

t1 = [[‘Valenciennes’, 24],[‘Lille’, 23],[‘Laon’, 31],[‘Arras’, 18]] t2 = [[‘Lille’, 62],[‘Arras’, 53],[‘Valenciennes’, 67],[‘Laon’, 48]]

for i in range(len(t1)) : for v in t2 : if v[0] == t1[i][0] : t1[i].append(v[1])

Réponses :

A- [[‘Valenciennes’, 67], [‘Lille’, 62], [‘Laon’, 48], [‘Arras’, 53]]

B- [[‘Valenciennes’, 24, 67], [‘Lille’, 23, 62], [‘Laon’, 31, 48], [‘Arras’, 18, 53]]

C- [[‘Arras’, 18, 53],[‘Laon’, 31, 48], [‘Lille’, 23, 62], [‘Valenciennes’, 24, 67]]

D- [[‘Valenciennes’, 67, 24], [‘Lille’, 62,23], [‘Laon’, 48, 31], [‘Arras’, 53, 18]]

Q103 - On écrit la fonction suivante :

def extreme(t, test) : m = t[0] for x in t : if test(x,m) : m = x return m

On dispose d’une liste L dont les éléments sont des couples (nom, note).

Par exemple :

L = [(‘Alice’, 17), (‘Barnabé’, 18),(‘Casimir’, 17), (‘Doriane’, 20),(‘Emilien’, 15), (‘Fabienne’, 16)]

On aimerait que l'appel de fonction extreme(L, test) renvoie un couple présentant la note maximale.

Quelle définition de la fonction test peut-on utiliser ?

Réponses :

A-

def test(a,b) : return a[0] < b[0]

B-

def test(a,b) : return a[0] > b[0]

C-

def test(a,b) : return a[1] < b[1]

D-

def test(a,b) : return a[1] > b[1]

Q104 - Une table d'un fichier client contient le nom, le prénom et l'identifiant des clients sous la forme :

clients = [("Dupont", "Paul", 1),("Durand", "Jacques", 2),("Dutronc", "Jean", 3),...]

En supposant que plusieurs clients se prénomment Jean, que vaut la liste x après l'exécution du code suivant ?

x = [] for i in range(len(clients)) : if clients[i][1] == "Jean" : x = clients[i]

Réponses :

A- Une liste de tuples des noms, prénoms et numéros de tous les clients prénommés Jean

B- Une liste des numéros de tous les clients prénommés Jean

C- Un tuple avec le nom, prénom et numéro du premier client prénommé Jean

D- Un tuple avec le nom, prénom et numéro du dernier client prénommé Jean

Q105 - Quelle est la valeur de x après exécution du programme ci-dessous ?

t = [[3,4,5,1],[33,6,1,2]] x = t[0][0] for i in range(len(t)) : for j in range(len(t[i])) : if x < t[i][j] : x = t[i][j]

Réponses :

A- 3

B- 5

C- 6

D- 33

Q106 - Quelle est la valeur de la variable table après exécution du programme Python suivant ?

table = [12, 43, 6, 22, 37] for i in range(len(table) - 1) : if table[i] > table[i+1] : table[i],table[i+1] = table[i+1], table[i]

Réponses :

A- [6, 12, 22, 37, 43]

B- [12, 6, 22, 37, 43]

C- [43, 12, 22, 37, 6]

D- [43, 37, 22, 12, 6]

Q107 - Laquelle de ces expressions a pour valeur la liste [[0,1,2],[3,4,5],[6,7,8]] ?

Réponses :

A- [[i+j for i in range(3)] for j in range(3)]

B- [[i]*3 for i in range(3)]*3

C- [[i+j*3 for i in range(3)] for j in range(3)]

D- [[i+j for i in range(3)] for j in range(3)]*3

Q108 - Si la variable note est définie par note = [“do”,“ré”,“mi”,“fa”,“sol”,“la”,“si”] alors :

Réponses :

A- l’index de “sol” est 5

B- l’index de note est 0

C- l’index de “si” est 7

D- l’index de “mi” est 2

Q109 - Un programme Python présente la ligne suivante

x = [“x1”, “x2”, “x3”]

Elle définit :

Réponses :

A- une liste de trois éléments

B- un tuple de trois éléments

C- une fonction acceptant trois paramètres

D- un dictionnaire associant la valeur x2 à la clé x1 d’indice x3

II. Dictionnaires

Q1 - On définit un dictionnaire :

d = {‘couleur’ : ‘vert’, ‘taille’ : 42, ‘marque’ : ‘le coq sportif’}

Quelle est la valeur de l’expression d.keys() ?

Réponses :

A- [‘couleur’, ‘taille’, ‘marque’]

B- [(‘couleur’, ‘vert’), (‘taille’, 42), (‘marque’, ‘le coq sportif’)]

C- [‘vert’, 42, ‘le coq sportif’]

D- [‘couleur’ : ‘vert’, ‘taille’ : 42, ‘marque’ : ‘le coq sportif’]

Q2 - Comment peut-on accéder à la valeur associée à une clé dans un dictionnaire ?

Réponses :

A- il faut parcourir le dictionnaire avec une boucle à la recherche de la clé

B- on peut y accéder directement à partir de la clé

C- on ne peut pas accéder à une valeur contenue dans un dictionnaire à partir d’une clé

D- il faut d’abord déchiffrer la clé pour accéder à un dictionnaire

Q3 - On définit le dictionnaire d = {‘a’ : 1, ‘b’ : 2, ‘c’ : 3, ‘z’ : 26}. Quelle expression permet de récupérer la valeur de la clé ‘z’ ?

Réponses :

A- d[4]

B- d[26]

C- d[z]

D- d[‘z’]

Q4 - Quel est le type de la variable billes définie par : billes = {‘vert’ : 6, ‘rouge’ : 15, ‘bleu’ : 11, ‘jaune’ : 2, ‘orange’ : 17 }

Réponses :

A- c’est une séquence

B- c’est une liste

C- c’est une liste de listes

D- c’est un dictionnaire

Q5 - Considérons le dictionnaire suivant : resultats = {‘Paul’ :5 , ‘Amina’ :1 , ‘Léon’ : 9 , ‘Benoit’ :3}

Quelle affirmation est correcte ?

Réponses :

A- resultats[‘Amina’] vaut 1

B- resultats[1] vaut ‘Amina’

C- ‘Paul’ est une valeur de ce dictionnaire

D- 9 est une clé de ce dictionnaire

Q6 - Après avoir défini : d = { ‘tigre’ : ‘félin’, ‘tortue’ : ‘reptile’, ‘renard’ : ‘canidé’ }

laquelle des quatre expressions suivantes est correcte ?

Réponses :

A- d[‘tortue’]

B- d[‘reptile’]

C- d[‘tortue’ : ‘reptile’]

D- d[1]

Q7 - On exécute le script suivant :

inventaire = {‘pommes’ : 430, ‘bananes’ : 312,‘oranges’ : 274, ‘poires’ : 137}

stock = 0 for fruit in inventaire.keys() : if fruit != ‘bananes’ : stock = stock + inventaire[fruit]

Que contient la variable stock à la fin de cette exécution ?

Réponses :

A- {430, 274, 137}

B- 312

C- 841

D- { ‘pommes’, ‘oranges’, ‘poires’ }

Q8 - On définit ainsi une liste P :

P = [{“nom” :“Turing”,“prénom” :“Alan”,“âge” :28},{“nom” :“Lovelace”,“prénom” :“Ada”,“âge” :27}]

Que fait alors l’instruction P[1][“âge”] = 25 ?

Réponses :

A- elle modifie la valeur de la clé âge du deuxième élément de la liste P

B- elle modifie la valeur de la clé âge du premier élément de la liste P

C- elle donne la longueur de la liste P

D- elle donne la longueur du premier élément de la liste P

Q9 - La variable sequence contient une liste de lettres, éventuellement répétées, choisies parmi ‘A’, ‘B’, ‘C’, ‘D’. On veut créer un dictionnaire effectifs associant à chaque lettre le nombre de fois qu’elle apparaît dans la liste sequence.

Par exemple si sequence contient [‘A’, ‘B’, ‘B’, ‘D’, ‘B’, ‘A’], effectifs doit contenir {‘A’ :2, ‘B’ :3, ‘C’ :0, ‘D’ :1}.

Parmi les scripts suivants, lequel réalise cet objectif ?

Réponses :

A-

effectifs = {‘A’ :0, ‘B’ :0, ‘C’ :0, ‘D’ :0} for lettre in sequence : effectifs[lettre] = effectifs[lettre] + 1

B-

effectifs = {} for lettre in sequence : effectifs[lettre] = effectifs[lettre] + 1

C-

effectifs = {‘A’ :0, ‘B’ :0, ‘C’ :0, ‘D’ :0} for lettre in effectifs.keys() : effectifs[lettre] = len([lettre in effectifs])

D-

effectifs = {} for lettre in effectifs.keys() : effectifs[lettre] = len([lettre in effectifs])

Q10 - On a défini

dico = { ‘a’ : (1,2,3), ‘b’ : (4,5,6) }

Quelle est la valeur de l’expression dico[‘a’][1] ?

Réponses :

A- 1

B- 2

C- (1,2,3)

D- cette expression est incorrecte, l’évaluer déclenche une erreur

Q11 - Quelle est la valeur affichée à l’exécution du programme Python suivant ?

ports = { ‘http’ : 80, ‘imap’ : 142, ‘smtp’ : 25 } ports[‘ftp’] = 21 print(ports[‘ftp’])

Réponses :

A- 3

B- 21

C- { ‘ftp’ : 21 }

D- Key not found

Q12 - On dispose du dictionnaire regions ci-dessous :

regions = {‘Mayotte’ : 376, ‘Pays de la Loire’ : 32082,‘La Réunion’ : 2504, ‘Grand Est’ : 57441,‘Martinique’ : 1128, ‘Corse’ : 8680,‘Bretagne’ : 27208, ‘Nouvelle-Aquitaine’ : 84036}

Parmi les instructions suivantes, laquelle permet d’ajouter une nouvelle région ?

Réponses :

A- INSERT “ ‘Hauts de France’ :31806” INTO regions

B- regions = dict([‘Hauts de France’] = 31806)

C- regions(‘Hauts de France’) = 31806

D- regions[‘Hauts de France’] = 31806

Q13 - On définit ainsi une liste P :

P = [{“nom” :“Turing”,“prénom” :“Alan”,“âge” :28},{“nom” :“Lovelace”,“prénom” :“Ada”,“âge” :27}]

Comment accéder à la chaîne de caractères “Alan” ?

Réponses :

A- P[0]

B- P[1]

C- P[0][“prénom”]

D- P[1][“prénom”]

Q14 - On exécute le script suivant :

def ajoute(stock,element,quantite) : if element in stock : stock[element] = stock[element] + quantite else : stock[element] = quantite

stock = { ‘clous’ : 14, ‘vis’ : 27, ‘boulons’ : 8, ‘écrous’ : 24 } ajoute(stock,‘vis’,5) ajoute(stock,‘chevilles’,3)

Quelle est la valeur de la variable stock à la fin de cette exécution ?

Réponses :

A- {‘clous’ : 14, ‘vis’ : 27, ‘boulons’ : 8, ‘écrous’ : 24}

B- {‘clous’ : 14, ‘vis’ : 32, ‘boulons’ : 8, ‘écrous’ : 24}

C- {‘clous’ : 14, ‘vis’ : 27, ‘boulons’ : 8, ‘écrous’ : 24, ‘chevilles’ : 3}

D- {‘clous’ : 14, ‘vis’ : 32, ‘boulons’ : 8, ‘écrous’ : 24, ‘chevilles’ : 3}

Q15 - On considère le code suivant :

D = { ‘a’ : ‘1’, ‘2’ : ‘a’, ‘b’ : ‘a’, ‘c’ : ‘3’}

Que vaut D[‘a’] à la fin de son exécution ?

Réponses :

A- ‘1’

B- 2

C- [‘2’, ‘b’]

D- [‘1’, ‘3’]

Q16 - On a défini un dictionnaire :

contacts = {‘Paul’ : ‘0601010182’, ‘Jacques’ : ‘0602413824’, ‘Claire’ : ‘0632451153’}

Quelle instruction écrire pour ajouter à ce dictionnaire un nouveau contact nommé Juliette avec le numéro de téléphone 0603040506 ?

Réponses :

A- ‘Juliette’ : ‘0603040506’

B- contacts.append(‘Juliette’ : ‘0603040506’)

C- contacts[‘Juliette’] = ‘0603040506’

D- contacts.append(‘Juliette’, ‘0603040506’)

Q17 - On considère le script suivant :

billes = {‘vert’ : 6, ‘rouge’ : 15, ‘bleu’ : 11, ‘jaune’ : 2, ‘orange’ : 17 } total = 0 for n in billes.XXXXXXX() : total = total + n

Par quoi faut-il remplacer XXXXXXX dans ce script pour qu'à la fin de son exécution la variable total contienne le nombre total de billes ?

Réponses :

A- keys

B- values

C- items

D- numbers

Q18 - On définit :

dico = {“Herve” : 15, “Kevin” :17, “Fatima” :16}

qui associe nom et âge de trois élèves.

Comment accéder à l'âge de Kevin ?

Réponses :

A- dico[1]

B- dico[Kevin]

C- dico[“Kevin”]

D- dico(“Kevin”)

Q19 - On exécute le code suivant :

placard = { ‘chemise’ : 3, ‘pantalon’ : 6, ‘tee shirt’ : 7 } placard[‘chaussette’] = 4 placard[‘chemise’] = 5

L = list(placard.values())

Quelle est la valeur de la variable L à l'issue de cette exécution ?

Réponses :

A- [3, 6, 7]

B- [3, 6, 7, 4]

C- [5, 6, 7]

D- [5, 6, 7, 4]

Q20 - Pour gérer certaines données EXIF de photographies, on a utilisé le code suivant pour stocker dans une liste L de dictionnaires quelques données :

L = [] L.append({‘marque’ : ‘Canon’, ‘modele’ : ‘EOS 7D’, ‘focale’ : ‘19mm’, ‘flash’ : False}) L.append({‘marque’ : ‘Nikon’, ‘modele’ : ‘CoolPix A1000’, ‘focale’ : ‘19mm’, ‘flash’ : True}) L.append({‘marque’ : ‘Sony’, ‘modele’ : ‘HK 350’, ‘focale’ : ‘24mm’, ‘flash’ : False}) L.append({‘marque’ : ‘Sony’, ‘modele’ : ‘HK 350’, ‘focale’ : ‘19mm’, ‘flash’ : True}) # …… # et ainsi de suite, d'autres informations ont été ajoutées # ……

On veut extraire de ces informations la liste Z des photographies obtenues avec un Canon ou un Nikon et une distance focale de 19 mm.

Quelle instruction permet de réaliser cette extraction ?

Réponses :

A-

Z = [p for p in L if (p[‘marque’] == ‘Canon’ or p[‘focale’] == ‘19mm’) and (p[‘marque’] == ‘Nikon’ or p[‘focale’] == ‘19mm’)]

B-

Z = [p for p in L if (p[‘marque’] == ‘Canon’ and p[‘focale’] == ‘19mm’) and (p[‘marque’] == ‘Nikon’ and p[‘focale’] == ‘19mm’)]

C-

Z = [p for p in L if (p[‘marque’] == ‘Canon’ or p[‘focale’] == ‘19mm’) or (p[‘marque’] == ‘Nikon’ or p[‘focale’] == ‘19mm’)]

D-

Z = [p for p in L if (p[‘marque’] == ‘Canon’ and p[‘focale’] == ‘19mm’) or (p[‘marque’] == ‘Nikon’ and p[‘focale’] == ‘19mm’)]

Q21 - On définit :

contacts = {‘Toto’ : ‘toto@nsi.fr’, ‘Chloé’ : ‘chloe@nsi.com’,‘Paul’ : ‘paul@nsi.net’, ‘Clémence’ : ‘clemence@nsi.org’ }

Parmi les propositions suivantes, laquelle est exacte ?

Réponses :

A- ‘Chloé’ est une valeur de la variable contacts

B- ‘Chloé’ est une clé de la variable contacts

C- ‘Chloé’ est un attribut de la variable contacts

D- ‘Chloé’ est un champ de la variable contacts

Q22 - On considère la table suivants :

t = [{‘type’ : ‘marteau’, ‘prix’ : 17, ‘quantité’ : 32},{‘type’ : ‘scie’, ‘prix’ : 24, ‘quantité’ : 3},{‘type’ : ‘tournevis’, ‘prix’ : 8, ‘quantité’ : 45}]

Quelle expression permet d’obtenir la quantié de scies ?

Réponses :

A- t[2][‘quantité’]

B- t[1][‘quantité’]

C- t[‘quantité’][1]

D- t[‘scies’][‘quantité’]

Q23 - On définit ainsi une liste t :

t = [{‘id’ :1, ‘age’ :23, ‘sejour’ :‘PEKIN’},{‘id’ :2, ‘age’ :27, ‘sejour’ :‘ISTANBUL’},{‘id’ :3, ‘age’ :53, ‘sejour’ :‘LONDRES’},{‘id’ :4, ‘age’ :41, ‘sejour’ :‘ISTANBUL’},{‘id’ :5, ‘age’ :62, ‘sejour’ :‘RIO’},{‘id’ :6, ‘age’ :28, ‘sejour’ :‘ALGER’}]

Quelle affirmation est correcte ?

Réponses :

A- t est une liste de listes

B- t est une liste de dictionnaires

C- t est un dictionnaire de listes

D- t est une liste de tuples

Q24 - On exécute le code suivant :

dict = {“alexandre” : 17, “mehdi” : 18, “jeanne” : 16,“charlotte” : 19, “celina” : 18, “noé” : 19}

def f(dic) : for cle, valeur in dic.items() : if valeur > 18 : return cle

Que renvoie l’appel f(dict) ?

Réponses :

A- 19

B- 19,19

C- “charlotte”

D- “charlotte”,“noé”

Q25 - On définit la variable suivante : lettres = {“a” : 1, “b” : 2, “c” : 3}.

Quelle est la valeur de l’expression list(lettres.keys()) ?

Réponses :

A- [a,b,c]

B- [1,2,3]

C- [“a”,“b”,“c”]

D- {“a” : 1, “b” : 2, “c” : 3}

Q26 - On exécute le script suivant :

notes = {“Paul” : 12, “Jean” : 16, “Clara” : 14, “Aïssa” : 18} t = list(notes.keys())

Quelle est la valeur de t à la fin de cette exécution ?

Réponses :

A- Paul

B- [“Paul”, ‘ “Jean”, “Clara”, ”’Aïssa’]

C- [12, 16, 14, 18]

D- [“Paul” : 12, “Jean” : 16, “Clara” : 14, “Aïssa” : 18]

Q27 - Par quelle expression remplacer les pointillés dans le programme Python suivant, pour que son exécution affiche le numéro de Dupond ?

repertoire = [{‘nom’ :‘Dupont’, ‘tel’ :‘5234’},{‘nom’ :‘Tournesol’, ‘tel’ :‘5248’}, {‘nom’ :‘Dupond’, ‘tel’ :‘3452’}] for i in range(len(repertoire)) : if …… : print(repertoire[i][‘tel’])

Réponses :

A- nom == ‘Dupond’

B- repertoire[‘nom’] == ‘Dupond’

C- repertoire[i] == ‘Dupond’

D- repertoire[i][‘nom’] == ‘Dupond’

Q28 - On définit :

T = [{‘fruit’ : ‘banane’, ‘nombre’ : 25}, {‘fruit’ : ‘orange’, ‘nombre’ : 124}, {‘fruit’ : ‘pomme’, ‘nombre’ : 75}, {‘fruit’ : ‘kiwi’, ‘nombre’ : 51}]

Quelle expression a-t-elle pour valeur le nombre de pommes ?

Réponses :

A- T[2][‘nombre’]

B- T[2,‘nombre’]

C- T[3][‘nombre’]

D- T[3,‘nombre’]

Q29 - Quelle expression Python permet d'accéder au numéro de téléphone de Tournesol, sachant que le répertoire a été défini par l'affectation suivante :

repertoire = [{'nom' :'Dupont', 'tel' :'5234'}, {'nom' :'Tournesol', 'tel' :'5248'}, {'nom' :'Dupond', 'tel' :'3452'}]

Réponses :

A- repertoire['Tournesol']

B- repertoire['tel'][1]

C- repertoire[1]['tel']

D- repertoire['Tournesol'][tel]

Q30 - On définit ainsi une liste t puis une liste r :

t = [{'id' :1, 'age' :23, 'sejour' :'PEKIN'},{'id' :2, 'age' :27, 'sejour' :'ISTANBUL'},{'id' :3, 'age' :53, 'sejour' :'LONDRES'},{'id' :4, 'age' :41, 'sejour' :'ISTANBUL'},{'id' :5, 'age' :62, 'sejour' :'RIO'},{'id' :6, 'age' :28, 'sejour' :'ALGER'}]

r = [c for c in t if c['age']>30 and c['sejour']=='ISTANBUL']

Combien la liste r contient-elle d'éléments ?

Réponses :

A- 0

B- 1

C- 2

D- 3

Q31 - On définit ainsi une liste t :

t = [{'id' :1, 'age' :23, 'sejour' :'PEKIN'},{'id' :2, 'age' :27, 'sejour' :'ISTANBUL'},{'id' :3, 'age' :53, 'sejour' :'LONDRES'},{'id' :4, 'age' :41, 'sejour' :'ISTANBUL'},{'id' :5, 'age' :62, 'sejour' :'RIO'},{'id' :6, 'age' :28, 'sejour' :'ALGER'}]

Quelle expression vaut-elle 'RIO' parmi les suivantes ?

Réponses :

A- t[4]['sejour']

B- t[5]['sejour']

C- t('id'=5)

D- t.['id'=5].['sejour']

Q32 - On considère des dictionnaires comme

{ 'nom' : 'Jérôme', 'NSI' : 16.2, 'maths' : 11.4, 'physique' : 13.0 }

pour retenir les notes d'un élève.

On définit :

def somme(notes) : return notes['NSI'] + notes['maths'] + notes['physique']

def plusPetit(n1, n2) : if n1['NSI'] < n2['NSI'] : return True if n1['NSI'] == n2['NSI'] : if somme(n1) < somme(n2) : return True elif somme(n1) == somme(n2) and n1['nom'] < n2['nom'] : return True return False

pour définir un ordre croissant sur ces dictionnaires.

Ranger dans l'ordre croissant les dictionnaires suivants :

n1 = { 'nom' : "Albert", 'NSI' : 12.3, 'maths' : 14.0, 'physique' : 8.7 } n2 = { 'nom' : "Béatrice", 'NSI' : 12.3, 'maths' : 11.0, 'physique' : 12.5 } n3 = { 'nom' : "Colin", 'NSI' : 12.3, 'maths' : 7.0, 'physique' : 15.7 } n4 = {

‘nom’ : “Daniel”, ‘NSI’ : 13.4, ‘maths’ : 9.0, ‘physique’ : 5.2 } n5 = { ‘nom’ : “Emilie”, ‘NSI’ : 16.1, ‘maths’ : 5.3, ‘physique’ : 14.4 }

Réponses :

A- n1, n2, n3, n4, n5

B- n1, n4, n2, n4, n5

C- n1, n3, n2, n4, n5

D- n5, n4, n2, n3, n1

Q33 - Par quoi faut-il remplacer les pointillés dans le script suivant :

relevé = [{‘matière’ :‘EPS’,‘moyenne’ :11}, {‘matière’ :‘Sciences’,‘moyenne’ :6}, {‘matière’ :‘LV1’,‘moyenne’ :14}, {‘matière’ :‘Histoire’,‘moyenne’ :9}, {‘matière’ :‘LV2’,‘moyenne’ :15}] a = …… b = …… for i in relevé : if i[a] > 10 : print(i[b])

pour qu’il affiche

EPS LV1 LV2

Réponses :

A-

a = ‘moyenne’ b = ‘matière’

B-

a = ‘matière b = ’moyenne’

C-

a = 0 b = 1

D-

a = 1 b = 0

Q34 - On définit :

stock = [{‘nom’ : ‘flageolets’, ‘quantité’ : 50, ‘prix’ : 5.68},{‘nom’ : ‘caviar’, ‘quantité’ : 0, ‘prix’ : 99.99},……,……,{‘nom’ : ‘biscuits’, ‘quantité’ : 100, ‘prix’ : 7.71}]

Quelle expression permet d’obtenir la liste des noms des produits effectivement présents dans le stock (c’est-à-dire ceux dont la quantité n’est pas nulle) ?

Réponses :

A- [‘nom’ for p in stock if ‘quantité’ != 0]

B- [p for p in stock if p[‘quantité’] != 0]

C- [p[‘nom’] for p in stock if ‘quantité’ != 0]

D- [p[‘nom’] for p in stock if p[‘quantité’] != 0]

Q35 - On considère le code suivant :

def clearfield(f) : for i in range(len(f)) : fiche[i][‘code’] = None return f

fiche = [{“nom” : “pierre”, “note” : 5.99, “code” : 125},{“nom” : “pol”, “note” : 2.99, “code” : 82},{“nom” : “jack”, “note” : 7.99, “code” : 135}]

Que renvoie clearfield(fiche) ?

Réponses :

A-

[{“nom” : “pierre”, “note” : 5.99, “code” : 125},{“nom” : “pol”, “note” : 2.99, “code” : 82},{“nom” : “jack”, “note” : 7.99, “code” : 135}]

B-

[{“nom” : “pierre”, “note” : None, “code” : 125},{“nom” : “pol”, “note” : None, “code” : 82},{“nom” : “jack”, “note” : None, “code” : 135}]

C-

[{“nom” : “pierre”, “note” : 5.99, “None” : 125},{“nom” : “pol”, “note” : 2.99, “None” : 82},{“nom” : “jack”, “note” : 7.99, “None” : 135}]

D-

[{“nom” : “pierre”, “note” : 5.99, “code” : None},{“nom” : “pol”, “note” : 2.99, “code” : None},{“nom” : “jack”, “note” : 7.99, “code” : None}]

Q36 - On a défini

repertoire = [{‘nom’ : ‘Francette’, ‘poste’ : 412},{‘nom’ : ‘Jeanne’, ‘poste’ : 222},{‘nom’ : ‘Éric’, ‘poste’ : 231}]

Quelle expression permet d’accéder au poste d’Éric ?

Réponses :

A- repertoire[2][‘poste’]

B- repertoire[‘poste’][2]

C- repertoire[‘Éric’][‘poste’]

D- repertoire[‘Éric’]