



Manh Cong NGUYEN (manhcuongeic@gmail.com)

Campagne : [SG] Python Campaign 2018 - #2

Langage(s) de programmation : Python3

Langage : Anglais

Date : 06/11/2018

SCORE

48%

740 / 1 550 pts

RANG

1

/ 1

DURÉE

0H41

/ 0H46



Python3



48%

(740 / 1 550)

Connaissance du langage



40%

(40 / 100)

Modélisation



38%

(300 / 800)

Résolution de problèmes



62%

(400 / 650)

Question 1: Iterate through dictionary



Python3



00:25 / 00:45



0 / 20 pts



Question

'd' is a dictionary, 'l' is a list. Which code is incorrect?



Réponse

- ☐ for k, v in d.items():
- ☐ for i, v in enumerate(l):
- ☒ for k, v in d:
- ☒ for k in d:



Résultat



Réponse incorrecte

Connaissance du langage ~~+20pts~~

Question 2: Strings mutability



Python3



00:09 / 00:20



0 / 20 pts



Question

Python strings are mutable



Réponse



False



True



Résultat



Réponse incorrecte

Connaissance du langage ~~+20pts~~

Question 3: List slices



Python3



00:25 / 00:25



0 / 20 pts



Question

What is the output?

```
l = [1, 2, 3, 4, 5, 6]  
print(l[5:3:-1])
```



Réponse



a. [6, 5]



b. [6, 5, 4]



c. [4, 3, 2]



d. [5, 4, 3]



Résultat



Réponse incorrecte

Connaissance du language ~~+20pts~~

Question 4: Classmethod



Python3



00:27 / 01:30



20 / 20 pts



Question

The @classmethod decorator alternates a method so that...



Réponse

- ☐ ...it could be called with classes as arguments
- ☒ **...when invoked as an object method, the calling object is ignored**
- ☐ ...it becomes a class-level method, that is impossible to invoke as an object method



Résultat



Réponse correcte

Connaissance du langage +20pts

Question 5: Tuple assign



Python3



00:06 / 00:25



20 / 20 pts

? Question

What is the output?

```
a, b = 1, 2
```

```
b, a = a, b
```

```
x = a, b
```

```
print(x)
```

✎ Réponse

☐ 1, 2

☐ 2, 1,

☐ (1, 2)

☒ (2, 1)

> Résultat



Réponse correcte

Connaissance du langage +20pts

Question 6: "Invert" a dictionary



Python3



02:00 / 02:00



0 / 50 pts

⚠ Le temps alloué à cette question s'est écoulé. La réponse du candidat a été automatiquement récupérée à la fin du décompte.

? Question

Implement a Python function that "inverts" a dictionary where keys are strings, and values are integers.

So that dictionary {'k1': v1, 'k2': v2} would become {v1: 'k1', v2: 'k2'}.

🔗 Réponse

```
1 # Python code below
2 # Use print("messages...") to debug your solution.
3
4 def invert_dict(d):
5     # Your code goes here
6     k={}
7     for i,j in d.items():
8         k[j]=i
9     inverted_d=k
10    return inverted_d
```

> Résultat



validate

Résolution de problèmes +50pts

Question 7: Vector multiplication



Python3



01:34 / 03:00



100 / 100 pts

? Question

Implement a function for vector multiplication.
Vectors are represented as Python lists.

(Vector multiplication: sum of the multiplication of individual elements at identical indexes.)

📝 Réponse

```
1 # Python code below
2 # Use print("messages...") to debug your solution.
3
4 def vect_mul(v1, v2):
5     # Your code goes here
6     result=0
7     for i in range(len(v1)):
8         result+=v1[i]*v2[i]
9     return result
```

> Résultat



validate

Résolution de problèmes +100pts

Question 8: Animal sounds



Python3



06:29 / 07:00



300 / 300 pts

? Question

Given an input string like:

"Well, sheep says beeeeeee and cat says miaaaaaaaaaaaaaaw - and a cow would shout moooooooooooooow".

Extract the animal sounds from the string into a list.

(Words that include character repetitions longer than a 2-length sequence, as shown in the example.)

📝 Réponse

```
1 # Python code below
2 # Use print("messages...") to debug your solution.
3 def get_words(s):
4     return s.split()
5 def find_sound(word):
6     d={}
7     for i in word:
8         d[i]=d.get(i,0)+1
9         if d[i]>len(word)/2 and len(word)>2:
10             return True
11     return False
12
13 def animal_sound(s):
14     # Your code goes here
15     sounds=[]
16     words=get_words(s)
17     for word in words:
18         if find_sound(word):
19             sounds.append(word)
20     return sounds
```

> Résultat



validate

Résolution de problèmes +300pts

Question 9: Reverse poem line



Python3



06:00 / 06:00



0 / 200 pts

⚠ Le temps alloué à cette question s'est écoulé. La réponse du candidat a été automatiquement récupérée à la fin du décompte.

? Question

Write a function that takes a poem and a line number as input, and prints a specified line reversed. See the format of a poem in the example code (ascii string with '\n' at the end of the lines).

📝 Réponse

```
1 # Python code below
2 # Use print("messages...") to debug your solution.
3 def split_poem(poem):
4     return poem.split('\n')
5 def reverse_line(st):
6     s=""
7     for i,c in enumerate(st):
8         s+=st[len(st)-i]
9     return s
10 def reverse_line(poem, n):
11     # Your code goes here
12     lines=split_poem(poem)
13     #print(lines[n-1])
14     #result=""
15     result=reverse_line(lines[n-1])
16     return result
```

> Résultat



validate

Résolution de problèmes ~~+200pts~~

Question 10: Sorted list class



Python3



07:00 / 07:00



0 / 200 pts

⚠ Le temps alloué à cette question s'est écoulé. La réponse du candidat a été automatiquement récupérée à la fin du décompte.

? Question

Implement a class that represents sorted lists (of basic orderable Python types, such as ascii characters or integers).

The test code defines expected behavior.

📝 Réponse

```
1 # Python code below
2 # Use print("messages...") to debug your solution.
3
4 # Implement the SortedList class
5
6 class SortedList:
7     def __init__(self, l):
8         self.l=l.sort()
9     def add(self,item):
10        self.l.append(item)
11        self.l.sort()
12    def concat(self, s2):
13        self.l.extend(s2.l)
14        self.l.sort()
```

> Résultat



validate

Modélisation ~~+200pts~~

Question 11: Binary search tree



Python3



09:00 / 09:00



0 / 300 pts

⚠ Le temps alloué à cette question s'est écoulé. La réponse du candidat a été automatiquement récupérée à la fin du décompte.

? Question

Implement a binary tree that is binary search tree. (Left descendants of a node are smaller, right descendants of a node are bigger or equal than the value held in the node.). Extending the tree has to keep this property.

Complete the sample code so that it corresponds to the test code.




🔑 Réponse

```
1 # Python code below
2 # Complete the definition of the class
3
4 class Node:
5
6     def __init__(self, n):
7         self.value = n
8         self.left = None
9         self.right = None
10
11     def add(self, n):
12         if self:
13             if self.value < n:
14                 add(self.right, n)
15             else:
16                 add(self.left, n)
17
18
19     def find(self, n):
20         if self.left == None and self.right == None and self.value != n:
21             return False
22         if self.value == n:
23             return True
24         if self.value < n:
25             return find(self.right, n)
26         else:
27             return find(self.left, n)
```

> Résultat

 validate
Modélisation ~~+300pts~~

Question 12: Logfile

 Python3  07:25 / 08:00  300 / 300 pts

? Question

We have a log file of the following format (separated by dash):

<timestamp>-<severity>-<message>

We would like to process it per record represented as a dictionary of the following format:

```
{'timestamp': '<timestamp>',  
'severity': '<severity>',  
'message': '<message>'}
```

What could be a most convenient solution?

Réponse

```
1 # Python code below
2 # Use print("messages...") to debug your solution.
3
4 # code for 'processlog' goes below
5 def get_lines(logs):
6     return logs.split('\n')
7
8 def split_line(line):
9     return line.split('-')
10
11 logs="""dkjkqjkd
12 qdkqjdfk
13 sqkjdkjqk
14 """
15
16 print (get_lines(logs))
17 def processlog(logs):
18     lines=get_lines(logs)
19     d=[]
20     for line in lines:
21         d_t={}
22         timestamp,severity,message=split_line(line)
23         d_t['timestamp']=timestamp
24         d_t['severity']=severity
25         d_t['message']=message
26         d.append(d_t)
27     return d
```

Résultat

 validate
Modélisation +300pts

Glossaire

Connaissance du langage

La mesure de cette compétence permet de déterminer l'expérience du candidat dans la pratique d'un langage de programmation. **Privilégiez cette compétence si, par exemple, vous recherchez un développeur qui devra être rapidement opérationnel.**

Design

Cette mesure fournit une indication sur la capacité du candidat à appliquer des solutions standard pour résoudre des problèmes récurrents. Un développeur ayant un bon niveau dans cette compétence augmentera la qualité (maintenabilité, évolutivité) de vos applications. Cette compétence ne dépend pas spécifiquement d'une technologie. **Privilégiez cette compétence si, par exemple, vous recherchez un développeur qui sera amené à travailler sur les briques qui structurent vos applications, à anticiper les besoins de demain pour développer des solutions pérennes.**

Résolution de problèmes

Cette compétence correspond aux aptitudes du candidat à comprendre et à structurer son raisonnement pour trouver des solutions à des problèmes complexes. Cette compétence ne dépend pas spécifiquement d'une technologie. **Privilégiez cette compétence si, par exemple, vos applications ont une composante technique importante (R&D, innovation).**

Fiabilité

La fiabilité caractérise la capacité du candidat à réaliser des solutions qui prennent en compte les cas particuliers. Plus cette compétence est élevée, plus vos applications sont robustes (moins de bugs).