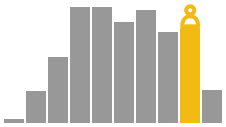
Manh Cuong NGUYEN (manhcuong.nguyen@softeam.fr)

Campagne : [SOFTEAM] Python - Level 1 Langage(s) de programmation : Python3 Langage : Français Date : 27/11/2018

SCORE	RANG	DURÉE	MEILLEUR QUE
80%	1	0H38	84%
980 / 1 220 pts	/ 2	/ 1H04	des développeurs



Python3

80%
(980 / 1 220)

Connaissance du language	90%	(360 / 400)
Fiabilité	70%	(184 / 264)
Modèlisation	67%	(80 / 120)
Résolution de problèmes	82%	(356 / 436)

Question 1: Instantiation d'objet

Python3 | 00:17 / 00:45 | 20 / 20 pts

Question

Comment crée-t-on une instance ***point*** de l'objet suivant ?

```
class Point():
    def __init__(self, x, y):
        self.x = x
        self.y = y
    def __eq__(self, other):
        return (self.x, self.y) == (other.x, other.y)
```

Réponse

- [ ] *point = new Point(x, y)*
- [ ] *point = Point(point, x, y)*
- [x] *point = Point(x, y)*

Résultat

Réponse correcte
Connaissance du language +20pts

Question 2: Boucle for

Python3 | 00:12 / 00:35 | 20 / 20 pts

Question

Comment peut-on itérer sur la liste suivante : ***arr = [1, 2, 3, 4, 5]*** ?

Réponse

- [x] *for n in arr:*
- [ ] *for n : arr:*
- [ ] *foreach n of arr:*

Résultat

Réponse correcte
Connaissance du language +20pts

Question 3: Héritage

Python3 | 00:50 / 02:00 | 50 / 50 pts

Question

Complétez la réponse afin que la classe B hérite de A.

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.

class A():
    def test(self):
        print("A")

    def __init__(self):
        self.test()


class B(A):
    def __init__(self):
        self.test()
```

Résultat

B hérite de A
Connaissance du language +50pts

Question 4: Declaration de fonction

Python3 | 00:17 / 00:30 | 20 / 20 pts

Question

Comment déclare-t-on une fonction ***name*** en Python?

Cochez toutes les propositions valides.

Réponse

- [x] *def name():*
- [ ] *name():*
- [ ] *function name():*
- [ ] *void name():*
- [ ] Il n'y a pas de fonctions en Python

Résultat

Réponse correcte
Connaissance du language +20pts

Question 5: Les type des éléments d'un tuple

Python3 00:14 / 00:30 20 / 20 pts

Question

Les éléments d'un ***tuple*** peuvent-ils être de types différents?

Réponse

- [x] Oui
- [ ] Non

Résultat

Réponse correcte
Connaissance du language +20pts

Question 6: Définition d'un dict

Python3 | 00:10 / 00:30 | 20 / 20 pts

Question

val = { '0' : 0, '1' : 1 , '2' : 2}

Quel est le type de ***val***?

Réponse

- [x] *dict*
- [ ] *list*
- [ ] *array*
- [ ] *map*
- [ ] *set*

Résultat

Réponse correcte
Connaissance du language +20pts

Question 7: Expression booléenne

Python3 00:22 / 00:30 20 / 20 pts

Question

Comment applique-t-on un ET entre deux booléens en Python ?

Réponse

- [ ] &&
- [x] *and*
- [ ] .
- [ ] ||

Résultat

Réponse correcte
Connaissance du language +20pts

Question 8: import os

Python3 | 00:16 / 00:35 | 20 / 20 pts

Question

Vous vous apprêtez à écrire du code Python permettant de gérer les fichiers dans un répertoire donné. Quelle(s) module(s) importerez-vous afin de naviguer dans le système de fichiers ?

Réponse

- [x] import os
- [ ] import fs
- [ ] import inspect
- [ ] import filesystem

Résultat

Réponse correcte
Connaissance du language +20pts

Question 9: Exceptions

Python3 | 00:12 / 00:35 | 20 / 20 pts

Question

Vous réalisez une bibliothèque en Python 3. Parmi ces options, laquelle privilégiez-vous pour traiter un comportement inattendu ?

Réponse

- [x]
```
raise Exception(
"Unexpected Behavior"
)
```

- [ ]
```
sys.
exit
(-1)
```

- [ ]
```
print("Unexpected
Behavior", file=sys.
stderr)
```

- [ ]
```
return
 False
```

Résultat

Réponse correcte
Modèlisation +20pts

Question 10: Threads

Python3 00:07 / 00:25 20 / 20 pts

Question

On peut multi-threader un programme Python.

Réponse

- [x] Vrai
- [ ] Faux

Résultat

Réponse correcte
Modèlisation +20pts

Question 11: Du désordre le plus grand gagne

Python3 | 00:55 / 05:00 | 100 / 100 pts

Question

Implémentez la fonction ***find_largest(numbers)*** afin qu'elle retourne le plus grand nombre dans la liste ***numbers***.

Note : La liste contient toujours au moins un nombre.

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.

def find_largest(numbers):
    # Your code goes here
    max=numbers[0]
    for i in numbers:
        if i>max:
            max=i
    return max
```

Résultat

- Fonctionne dans des cas simples
  Résolution de problèmes +32pts
  - Fonctionne quand le tableau contient seulement Integer.MIN_VALUE
    Fiabilité +58pts
  - Fonctionne quand le plus grand élément est à la position 0
    Fiabilité +5pts
  - Fonctionne quand le plus grand élément est à la fin du tableau
    Fiabilité +5pts

Question 12: Chaînes de caractères égales

Python3 | 00:54 / 02:30 | 50 / 50 pts

Question

is_foo(param) devrait retourner ***True*** si ***param*** est égal à la chaine ***"foo"***, sinon elle devrait retourner ***False***.

Implémentez la fonction ***is_foo(param)***.

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.

def is_foo(param):
    # Your code goes here
    if param=="foo":
        return True
    return False
```

Résultat

Utilisation de '=='
Résolution de problèmes +17pts

Fonctionne si param est None
Fiabilité +33pts

Question 13: append()

Python3 | 00:11 / 00:30 | 40 / 40 pts

Question

Parmis les propositions suivantes, lesquelles permettent de rajouter ***5*** à la liste ***arr = [1,2,3,4] ?***

Cochez toutes les propositions valides.

Réponse

- [ ] *arr.add(5)*
- [x] *arr.append(5)*
- [ ] *arr.push(5)*
- [ ] *arr += 5*

Résultat

Réponse correcte
Connaissance du language +40pts

Question 14: Tuple vs Liste

Python3 00:21 / 01:00 40 / 40 pts

Question

Quelle est la différence entre les types ***tuple*** et ***list*** ?

Réponse

- [ ] ***list*** est ordonnée, ***tuple*** ne l'est pas.
- [ ] ***list*** peut contenir des doublons, ***tuple*** contient des valeurs uniques.
- [ ] ***tuple*** peut contenir des valeurs de différents type, ***list*** ne peut pas.
- [x] ***tuple*** **est immutable,** ***list*** **est mutable.**

Résultat

Réponse correcte
Connaissance du language +40pts

Question 15: Existence d'une clef dans un dict

Python3 | 00:32 / 01:00 | 40 / 40 pts

Question

Quel (ou quelles) instructions permet(tent) de vérifier si la clef ***"Bob"*** est présente dans le dictionnaire ***annuaire*** ?

Réponse

- [x] **"Bob" in annuaire**
- [ ] annuaire["Bob"] is not None
- [ ] annuaire["Bob"] != None
- [ ] annuaire.Bob != None
- [ ] annuaire.contains("Bob")

Résultat

Réponse correcte
Connaissance du language +40pts

Question 16: Ordre d'éxecution

Python3 | 00:58 / 01:00 | 0 / 40 pts

Question

Le code ci-dessous se trouve dans un fichier ***file.py***. En lançant la commande ***python3 file.py***, dans quel ordre les différents blocs s'exécuteront-ils ?

```
#code block A

def main():
    #code block B

if __name__ == '__main__':
    main()

#code block C
```

Réponse

- [ ] A puis B puis C
- [ ] seulement B
- [ ] A puis B
- [x] A puis C puis B
- [ ] A puis C

Résultat

Réponse incorrecte
Connaissance du language ~~+40pts~~

Question 17: Moyenne

Python3 02:30 / 02:30 100 / 100 pts

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

Question

Implémentez la fonction ***average(table)*** .

La fonction doit renvoyer la valeur moyenne du tableau ***table*** donné en paramètre. ***table*** est toujours un tableau défini, et ne contient que des nombres.

average doit retourner ***0*** si ***table*** est vide.

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.

def average(table):
    # Your code goes here
    sum=0
    if len(table)==0:
        return 0
    else:
       for i in table:
           sum+=i


    return sum*1.0/len(table)
    return 0
```

Résultat

La solution calcule des moyennes
Connaissance du language +50pts

La solution fonctionne avec un tableau vide
Fiabilité +25pts

La solution fonctionne avec un large jeu de données
Fiabilité +25pts

Question 18: Tableau vide ou None ?

Python3 00:29 / 00:30 40 / 40 pts

Question

Vous écrivez une fonction qui retourne une liste de fichiers appartenant à un répertoire. Parmi ces options, laquelle est à privilégier si le répertoire est vide ?

Réponse

- [x] **La fonction devrait retourner une liste vide.**
- [ ] La fonction devrait retourner ***None***.
- [ ] La fonction devrait lever une exception.

Résultat

Réponse correcte
Modèlisation +40pts

Question 19: super()

Python3 | 00:27 / 00:45 | 0 / 40 pts

Question

Considérez le code Python 3 ci-dessous.

```
class A:
    def __init__(self, text):
        self.text = text

class B(A):
    def __init__(self):
        #TODO
```

Parmi ces intructions, laquelle doit être utilisée pour remplacer le #TODO?

Réponse

- [ ]
```
super().
__init__
('hello'
)
```

- [x]
```
super(
'hello'
)
```

Résultat

Réponse incorrecte
Modèlisation ~~+40pts~~

Question 20: Rendu de monnaie

Python3 26:15 / 40:00 240 / 400 pts

Question

Les supermarchés s'équipent de plus en plus de caisses automatiques. La plupart de ces caisses n' acceptent que le paiement par carte bancaire bien qu'une part non négligeable de consommateurs paye encore en espèces (avec des billets et des pièces).

Une des problématiques rencontrées avec le paiement en espèces est le rendu de monnaie : comment rendre une somme donnée de façon optimale, c'est-à-dire avec le nombre minimal de pièces et billets ? C'est un problème qui se pose à chacun de nous quotidiennement, à fortiori aux caisses automatiques.

Dans cet exercice, on vous demande d'essayer de trouver une solution optimale pour rendre la monnaie dans un cas précis : quand une caisse automatique ne contient que des pièces de 2€, des billets de 5€ et de 10€.

Pour simplifier le problème, nous considérerons que toutes ces pièces et billets sont disponibles en quantité illimitée.

Voici quelques exemples de rendu de monnaie :
Monnaie à rendre Solutions possibles Solution optimale 1 Impossible Impossible 6 2 + 2 + 2 2 + 2 + 2 10 2 + 2 + 2 + 2 + 2 5 + 5 10 10 9223372036854775807 ... (10 * 922337203685477580) + 5 + 2

Le rendu de monnaie est exprimé par un dictionnaire avec trois clés : ***two***, ***five*** et ***ten*** qui donnent respectivement le nombre de pièces de 2€, de billets de 5€ et de billets de 10€.

Par exemple, si on reprend l'exemple n°2 du tableau (6€), on devrait obtenir le dictionnaire suivant :

```
{
    'two': 3, # 3 pièces de 2€
    'five': 0 # aucun billet de 5€
    'ten': 0 # aucun billet de 10€
}
```

Implémentez la fonction ***change(cash)*** qui retourne un dictionnaire dont les attributs two, five, ten représentent la monnaie à rendre.

S'il est impossible de rendre la monnaie (comme dans l'exemple n°1), retournez ***None***.

Pour obtenir un maximum de points votre solution devra toujours rendre la monnaie quand c'est possible et avec le nombre minimal de pièces et billets.

Contraintes : 0 < cash < 9007199254740991

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.

def change(cash):
    # Your code goes here
    find=False
    if cash<=1:
        return "None"
    else:
        sum=0
        sum_min=int(cash)
        i_index,j_index,k_index=0,0,0
        for i in range(int(cash/2)+1):
            for j in range(int(cash/5)+1):
                for k in range(int(cash/10)+1):
                    sum=i*2+j*5+k*10
                    if sum==cash:
                        if sum_min>i+j+k:
                            i_index,j_index,k_index=i,j,k
                            sum_min=i_index+j_index+k_index
                            find=True
        if find==True:
            return{'two':i_index,
                   'five':j_index,
                   'ten':k_index
                }
        else:
            return "None"


```

Résultat

- La monnaie est correcte pour une somme de 10€
  Résolution de problèmes +40pts
  - La monnaie est optimale pour une somme de 10€ (1*10)
    Résolution de problèmes +40pts
  - ***None*** est retourné quand la somme vaut 1
    Fiabilité ~~+40pts~~
  - Le programme rend correctement la monnaie quand la somme vaut 31
    Résolution de problèmes +40pts
  - La monnaie est optimale pour une somme de 31€ (2*10 +5 + 3*2)
    Résolution de problèmes +40pts
  - ***None*** est retourné quand la somme vaut 3
    Fiabilité ~~+40pts~~
  - Le programme rend correctement la monnaie quand la somme vaut 8 (4 * 2)
    Résolution de problèmes +40pts
  - La monnaie est optimale pour une somme de 8€ (4*2)
    Résolution de problèmes +40pts
  - Résultat correct et dans les temps avec 9007199254740991€
    Résolution de problèmes ~~+40pts~~
  - La monnaie est optimale avec 9007199254740991€
    Résolution de problèmes ~~+40pts~~

Question 21: Simple expression booléenne

Python3 01:13 / 02:00 100 / 100 pts

Question

is_bool(i, j) devrait retourner ***True*** si un des arguments est égal à 1 ou si leur somme est égale à 1.

Par exemple :
is_bool(1, 5) retourne ***True***
is_bool(2, 3) retourne ***False***
is_bool(-3, 4) retourne ***True***

Réponse

```python
# Python code below
# Use print("messages...") to debug your solution.

def is_bool(i, j):
    # Your code goes here
    if i==1 or j==1 or i+j==1:
        return True
    return False
```

Résultat

Retourne True si i ou j est égal à 1, sinon False
Résolution de problèmes +67pts

Retourne True si i+j est égal à 1
Fiabilité +33pts

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).