

Frédéric  
LURET



# Python

## Banque de questions

Version du 15 décembre 2024

**0 Table des matières**

1 Site 3

2

**1 Site 3**[Lien vers le site d'origine](#)**Question 1**

Définir une fonction nommée `is_two`. Elle doit accepter une entrée et retourner `True` si l'entrée passée est soit le nombre 2, soit la chaîne 2, `False` sinon.

**Code python :**

```
1 def is_two(x):
2     if x == 2 or x == "2":
3         return True
4     else:
5         return False
6
7
8 print(is_two(2))
9 print(is_two("2"))
10 print(is_two(3))
11 print(is_two("3"))
```

q200.py

**Question 2**

Définir une fonction nommée `is_vowel`. Elle doit renvoyer `True` si la chaîne passée est une voyelle, `False` sinon.

**Code python :**



```
1 def is_vowel(vowel):
2     vowels = "aeiouAEIOU"
3     if vowel in vowels:
4         return True
5     else:
6         return False
7
8
9 print(is_vowel("a"))
10 print(is_vowel("b"))
11 print(is_vowel("A"))
12 print(is_vowel("B"))
```

q201.py

### Question 3

Définir une fonction nommée `is_consonant`. Elle doit retourner `True` si la chaîne passée est une consonne, `False` sinon. Utilisez votre fonction `is_vowel`

Code python :

```
1 def is_consonant(consonant):
2     vowels = "aeiouAEIOU"
3     if consonant not in vowels:
4         return True
5     else:
6         return False
7
8 print(is_consonant("a"))
9 print(is_consonant("b"))
10 print(is_consonant("A"))
11 print(is_consonant("B"))
```

q202.py

### Question 4

Définir une fonction qui accepte une chaîne de caractères qui est un mot. La fonction doit mettre en majuscule la première lettre du mot si celui-ci commence par une consonne.

Code python :



```
1 def cap_if_consonant(x):
2     vowels = "aeiouAEIOU"
3     for letter in vowels:
4         if x[0] in vowels:
5             return x
6         else:
7             return x.capitalize()
8
9
10 print(cap_if_consonant("great job!"))
```

q203.py

### Question 5

Définissez une fonction nommée `calculate_tip`. Elle doit accepter un pourcentage de pourboire (un nombre entre 0 et 1) et le total de l'addition, et renvoyer le montant du pourboire.

Code python :

```
1 def calculate_tip(x, y):
2     bill_total = x
3     tip_percentage = y
4     tip = bill_total * tip_percentage
5     return f"${tip}"
6
7
8 print(calculate_tip(10, 0.15))
```

q204.py

### Question 6

Définissez une fonction nommée `apply_discount`. Elle doit accepter un prix d'origine et un pourcentage de remise, et renvoyer le prix après la remise.

Code python :

```
1 def apply_discount(x, y):
2     original_price = x
3     discount_percentage = y
4     total = original_price - (original_price * discount_percentage)
5     return f"${total}"
6
7
8 print(apply_discount(10, 0.2))
```

q205.py

**Question 7**

Définissez une fonction nommée `handle_commas`. Elle doit accepter en entrée une chaîne de caractères qui est un nombre contenant des virgules, et retourner un nombre en sortie.

Code python :

```
1 def handle_commas(x):
2     no_commas = int(x.replace(",", ""))
3     return no_commas
4
5
6 print(handle_commas("1,000,000"))
```

q206.py

**Question 8**

Définissez une fonction nommée `get_letter_grade`. Elle doit accepter un nombre et retourner la lettre associée à ce nombre (A-F).

Code python :

```
1 def get_letter_grade(grade):
2     if grade >= 100:
3         print("Please enter a grade between 0-100")
4     elif grade >= 90:
5         print(f"{grade} is an A")
6     elif grade >= 80:
7         print(f"{grade} is a B")
8     elif grade >= 70:
9         print(f"{grade} is a C")
10    elif grade >= 60:
11        print(f"{grade} is a D")
12    elif grade <= 59 and grade >= 0:
13        print(f"{grade} is an F")
14    else:
15        print("No negative numbers")
16
17
18 get_letter_grade(95)
19 get_letter_grade(85)
20 get_letter_grade(75)
21 get_letter_grade(65)
22 get_letter_grade(55)
23 get_letter_grade(0)
24 get_letter_grade(105)
25 get_letter_grade(-5)
```

q207.py

**Question 9**

Définissez une fonction nommée `remove_vowels` qui accepte une chaîne et renvoie une chaîne dont toutes les voyelles ont été supprimées.

Code python :

```
1 def remove_vowels(string):
2     vowels = ("a", "e", "i", "o", "u")
3     for x in string.lower():
4         if x in vowels:
5             string = string.replace(x, "")
6     return string
7
8
9 print(remove_vowels("Hello Codeup!"))
```

q208.py

**Question 10**

Définissez une fonction nommée `normalize_name`. Elle doit accepter une chaîne et retourner un identifiant python valide, c'est-à-dire :

- tout ce qui n'est pas un identifiant python valide doit être supprimé
- les espaces blancs de début et de fin doivent être supprimés
- tout doit être en minuscules
- les espaces doivent être remplacés par des traits de soulignement

par exemple :

- Nom deviendra nom
- Prénom deviendra prénom
- Completed deviendra completed

Code python :



```
1 def normalize_name(text):
2     new_string = text.strip("0123456789 ").lower().replace(" ", "_")
3     output = ""
4     if new_string.isidentifier():
5         print(new_string, "is a valid identifier")
6         for i in new_string:
7             if i not in "!@#$%^&*()+=-[]{}\\|?<.>`~":
8                 output += i
9     else:
10        print(new_string, "is NOT a valid identifier")
11    return print(output)
12
13
14 normalize_name("one+one")
15 normalize_name(" Dani Bojado ")
```

q209.py

### Question 11

Écrivez une fonction nommée `cumulative_sum` qui accepte une liste de nombres et renvoie une liste qui est la somme cumulative des nombres de la liste.

- `cumulative_sum([1, 1, 1])` renvoie `[1, 2, 3]`
- `cumulative_sum([1, 2, 3, 4])` renvoie `[1, 3, 6, 10]`

Code python :

```
1 def cumulative_sum(num_list):
2     return [sum(num_list[: i + 1]) for i in range(len(num_list))]
3
4
5 print(cumulative_sum([1, 1, 1]))
6 print(cumulative_sum([1, 2, 3, 4]))
```

q210.py

### Question 12

Soit les deux listes suivantes :

`fruits = ['mango', 'kiwi', 'strawberry', 'guava', 'pineapple', 'mandarin orange']`

`numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4, -2, 5, -9]`

Réécrire l'exemple de code ci-dessus en utilisant la syntaxe de compréhension de liste. Créez une variable nommée `uppercased_fruits` pour contenir la sortie de la compréhension de liste. La sortie devrait être `['MANGO', 'KIWI', etc...]`.

Code python :



```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 uppercased_fruits = [fruit.upper() for fruit in fruits]  
5 print(uppercased_fruits)
```

q211.py

### Question 13

Créer une variable nommée `capitalized_fruits` et utiliser la syntaxe de compréhension de liste pour produire des résultats comme ['Mango', 'Kiwi', 'Strawberry', etc...].

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 capitalized_fruits = [fruit.capitalize() for fruit in fruits]  
5 capitalized_fruits
```

q212.py

### Question 14

Utilisez une compréhension de liste pour créer une variable nommée `fruits_avec_plus_de_deux_vo`.  
Astuce : Vous aurez besoin d'un moyen de vérifier si quelque chose est une voyelle.

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 fruits_with_more_than_two_vowels = [fruit for fruit in fruits if (  
5     fruit.count("a") +  
6     fruit.count("e") +  
7     fruit.count("i") +  
8     fruit.count("o") +  
9     fruit.count("u")) > 2]  
10  
11 print(fruits_with_more_than_two_vowels)
```

q213.py



**Question 15**

Créer une variable nommée `fruits_avec_seulement_deux_voyelles`.  
Le résultat devrait être `['mangue', 'kiwi', 'fraise']`.

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
  ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
  ↪ -2, 5, -9]  
3  
4 fruits_with_only_two_vowels = [  
5     fruit  
6     for fruit in fruits  
7     if (  
8         fruit.count("a")  
9         + fruit.count("e")  
10        + fruit.count("i")  
11        + fruit.count("o")  
12        + fruit.count("u")  
13    )  
14    == 2  
15 ]  
16  
17 print(fruits_with_only_two_vowels)
```

q214.py

**Question 16**

Faire une liste qui contient chaque fruit avec plus de 5 caractères

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
  ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
  ↪ -2, 5, -9]  
3  
4 print([fruit for fruit in fruits if (len(fruit) > 5)])
```

q215.py

**Question 17**

Faire une liste qui contient chaque fruit avec exactement 5 caractères

Code python :



```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
  ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
  ↪ -2, 5, -9]  
3  
4 print([fruit for fruit in fruits if (len(fruit) == 5)])
```

q216.py

### Question 18

Faire une liste qui contient des fruits qui ont moins de 5 caractères

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
  ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
  ↪ -2, 5, -9]  
3  
4 print([fruit for fruit in fruits if (len(fruit) < 5)])
```

q217.py

### Question 19

Faites une liste contenant le nombre de caractères de chaque fruit. Les résultats seraient 5, 4, 10, etc...

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
  ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
  ↪ -2, 5, -9]  
3  
4 print([len(fruit) for fruit in fruits])
```

q218.py

### Question 20

Créez une variable nommée fruits\_avec\_lettre\_a qui contient une liste des seuls fruits contenant la lettre "a"

Code python :



```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 fruits_with_letter_a = [fruit for fruit in fruits if "a" in fruit]  
5 print(fruits_with_letter_a)
```

q219.py

### Question 21

Créer une variable nommée `even_numbers` qui ne contiendra que les nombres pairs

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 even_numbers = [number for number in numbers if number % 2 == 0]  
5 print(even_numbers)
```

q220.py

### Question 22

Créer une variable nommée `nombres_impairs` qui ne contient que les nombres impairs

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 odd_numbers = [number for number in numbers if number % 2 != 0]  
5 print(odd_numbers)
```

q221.py

### Question 23

Créer une variable nommée `nombres_positifs` qui ne contient que les nombres positifs

Code python :



```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 positive_numbers = [number for number in numbers if number > 0]  
5 print(positive_numbers)
```

q222.py

### Question 24

Créer une variable nommée `nombre_négatifs` qui ne contient que les nombres négatifs

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 negative_numbers = [number for number in numbers if number < 0]  
5 print(negative_numbers)
```

q223.py

### Question 25

Utiliser une compréhension de liste avec un conditionnel afin de produire une liste de nombres avec 2 chiffres ou plus

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 print([number for number in numbers if number > 9 or number < -9])
```

q224.py

### Question 26

Créez une variable nommée `numbers_squared` qui contient la liste des nombres avec chaque élément au carré. La sortie est `[4, 9, 16, etc...]`

**Code python :**

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 numbers_squared = [number ** 2 for number in numbers]  
5 print(numbers_squared)
```

q225.py

**Question 27**

Créez une variable nommée `nombre_impairs_négatifs` qui ne contient que les nombres qui sont à la fois impairs et négatifs.

**Code python :**

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 odd_negative_numbers = [number for number in numbers if number < 0 and  
    ↪ number % 2 != 0]  
5 print(odd_negative_numbers)
```

q226.py

**Question 28**

Créez une variable nommée `nombre_plus_5`. Dans cette variable, renvoyez une liste contenant chaque nombre plus cinq.

**Code python :**

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 numbers_plus_5 = [number + 5 for number in numbers]  
5 print(numbers_plus_5)
```

q227.py

**Question 29**

Créez une variable nommée `"primes"` qui est une liste contenant les nombres premiers de la liste des nombres. \*Astuce : vous pouvez créer ou trouver une fonction d'aide



qui détermine si un nombre donné est premier ou non.

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
  ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
  ↪ -2, 5, -9]  
3  
4  
5 def is_prime(num):  
6     prime_check = False  
7     if num > 1:  
8         for i in range(2, num):  
9             if (num % i) == 0:  
10                prime_check = False  
11                break  
12        else:  
13            prime_check = True  
14    return prime_check  
15  
16  
17 primes = [number for number in numbers if is_prime(number)]  
18 print(primes)
```

q228.py