Frédéric
LURET

Python
Banque de questions

Version du 15 décembre 2024

Université de TOURS

0 Table des matières

1 Site 7

Lien vers le site d'origine

Question 1

Trouvez tous les nombres compris entre 1 et 1000 qui sont divisibles par 7.

Code python :

```python
div7 = [n for n in range(1,1000) if n % 7 == 0]
print(div7)
```

q135.py

Question 2

Trouvez tous les nombres de 1-1000 qui contiennent un 3.

Code python :

```python
three = [n for n in range(0,1000) if '3' in str(n)]
print(three)

x = [i for i in range(1, 1001) if str(i).find("3") != -1]
y = [i for i in range(1, 1001) if str(i).count("3") > 0]
z = [i for i in range(1, 1001) if '3' in str(i)]

print("x ->", x, len(x), "\n")
print("y ->", y, len(y), "\n")
print("z ->", z, len(z))
```

q136.py

Question 3

Compter le nombre d'espaces dans une chaine.

Code python :

```python
some_string = "the slow solid squid swam sumptuously through the slimy swamp"
spaces = [s for s in some_string if s == " "]
print(len(spaces))

res = sum([1 for x in some_string if x == " "])
```

q137.py

Question 4

Créer une liste de toutes les consonnes de la chaîne "Les Yaks jaunes aiment crier et bailler et hier ils ont jodlé en mangeant des ignames yuky".

Code python :

```python
sentence = "Yellow Yaks like yelling and yawning and yesturday they yodled while eating yuky yams"
result = [letter for letter in sentence if letter not in 'a,e,i,o,u, " "']
print(result)

stringex4 = "Yellow Yaks like yelling and yawning and yesturday they yodled while eating yuky yams"
print(list(a for a in stringex4 if a not in ("a", "e", "i", "o", "u", " ")))

import string

stringex4 = "Yellow Yaks like yelling and yawning and yesturday they yodled while eating yuky yams, special characters: * ' and ()/"
print(
    list(
        a
        for a in stringex4
        if a not in ("a", "e", "i", "o", "u", " ")
        and (a in list(string.ascii_lowercase) or a in list(string.ascii_uppercase))
    )
)

sentence = "Yellow Yaks like yelling and yawning and yesturday they yodled while eating yuky yams"
consonant_list = [letter for letter in sentence if letter not in 'a, e, i, o, u, " "']
print(consonant_list)

string = "Yellow Yaks like yelling and yawning and yesturday they yodled while eating yuky yams, special characters: * ' and ()/"
print(
    [
        char
        for char in string
        if ("A" <= char <= "Z" or "a" <= char <= "z") and char not in "aeiouAEIOU"
    ]
)

# I would just propose to transforme the list in a set to have a unique count of each different consonnant :

some_string = "Yellow Yaks like yelling and yawning and yesturday they yodled while eating yuky yams"
voyelles = ["a", "e", "i", "o", "u", "y", " "]
consonnant = [c for c in some_string.lower() if c not in voyelles]
print(set(consonnant))

vowels_space = ["a", "e", "i", "o", "u", " "]
sentence = "Yellow Yaks like yelling and yawning and yesturday they yodled while eating yuky yams"
my_list = [letter for letter in sentence.lower() if letter not in vowels_space]

sentence = "Yellow Yaks like yelling and yawning and yesturday they yodled while eating yuky yams"
consonant = [
    consonant
    for consonant in sentence
    if consonant.lower() not in "aeiou" and consonant != " "
]
print(f"{consonant}")

vowels = "aeıioöuüAEIİOÖUÜ"
text = "Yellow Yaks like yelling and yawning and yesturday they yodled while eating yuky yams"
consontants = [i for i in text if i not in vowels]
print(consontants)
```

q138.py

Question 5

Obtenir l'indice et la valeur sous forme de tuple pour les éléments de la liste ["hi", 4, 8.99, 'apple', ('t,b','n')]. Le résultat ressemblerait à [(index, valeur), (index, valeur)].

Code python :

```python
items = ["hi", 4, 8.99, 'apple', ('t,b','n')]
result = [(index, item) for index, item in enumerate(items)]
print(result)

items=["hi", 4, 8.99, 'apple', ('t,b','n')]
result=[n for n in enumerate(items)]
print(result)

mylist = ["hi", 4, 8.99, "apple", ("t,b","n")]
result = [(tuple([index, mylist[index]])) for index in range(len(mylist))]
print(result)



strr=["hi", 4, 8.99, 'apple', ('t,b','n')]
l=[(strr.index(x),x) for x in strr]
print(l)
```

q139.py

Question 6

Trouver les nombres communs à deux listes (sans utiliser de tuple ou d'ensemble) list_a = [1, 2, 3, 4], list_b = [2, 3, 4, 5]

Code python :

```python
list_a = [1, 2, 3, 4, 3, 4]
list_b = [2, 3, 4, 5, 1]
common = [a for a in list_a if a in list_b]
print(common)

list_a = 1, 2, 3, 4, 3, 4
list_b = 2, 3, 4, 5
my_list = [element_b for element_b in list_b for element_a in list_a if element_b == element_a]
print(my_list)

common_numbers = []
def f(x):
    common_numbers.append(x)
    return x

list_a = 1, 2, 3, 4, 3, 4
list_b = 2, 3, 4, 5,
my_list = [f(element_b) for element_b in list_b for element_a in list_a if element_b == element_a and element_b not in common_numbers]
print(my_list)

list_a = [1, 2, 3, 4, 3, 4]
list_b = [2, 3, 4, 5, 1]
common = []
[common.append(a) for a in list_a if a in list_b and a not in common]
print(common)
```

q140.py

Question 7

Dans une phrase comme "En 1984, il y a eu 13 cas de manifestations ayant rassemblé plus de 1 000 personnes", il ne faut retenir que les chiffres. Le résultat est une liste de nombres comme [3,4,5].

Code python :

```python
sentence = 'In 1984 there were 13 instances of a protest with over 1000 people attending'
words = sentence.split()
result = [number for number in words if not number.isalpha() ]
print(result)

sentence = 'In 1984 there were 13 instances of a protest with over 1000 people attending'
words = sentence.split()
result = [number for number in words if number.isnumeric()]
print(result)

state = 'In 1984 there were 13 instances of a protest with over 1000 people attending'
st_list = [word for word in state.split() if word.isdigit()]
print(st_list)

import re

[i for i in "In 1984 there were 13 instances of a protest with over 1000 people attending".split() if re.match("[0-9]", i)]



import re

sentence = 'In 1984 there were 13 instances of a protest with over 1000 people attending'
w_in_sentence = sentence.split()

pattern = r'[0-9]'

print([n for n in sentence if re.match(pattern, n)])

```

q141.py

Question 8

Étant donné numbers = range(20), produisez une liste contenant le mot "even" si un des nombres est pair, et le mot "odd" si le nombre est impair. Le résultat ressemblerait à ['odd', 'odd', 'even'].

Code python :

```python
result = ["even" if n % 2 == 0 else "odd" for n in range(20)]
print(result)

"""
Let's see the for loop and break out the syntax of the list comprehension
"""
result = []
for n in range(20):
    if n % 2 == 0:
        result.append("even")
    else:
        result.append("odd")

"""
List comprehension
[expression for item in list]
expression = "'even' if n %2 == 0 else 'odd'"
for item in list = "for n in range(20)"
"""

[['even','odd'][x%2] for x in numbers]

res = ['odd' if n%2 else 'even' for n in range(20)]
```

q142.py

Question 9

Produisez une liste de tuples composée uniquement des nombres correspondants dans ces listes list_a = [1, 2, 3, 4, 5, 6, 7, 8, 9], list_b = [2, 7, 1, 12]. Le résultat ressemblerait à (4,4), (12,12)

Code python :

```python
list_a = [1, 2, 3,4,5,6,7,8,9]
list_b = [2, 7, 1, 12]

result = [(a, b) for a in list_a for b in list_b if a == b]
print(result)

print([(i,i) for i in list_a if i in list_b])
```

q143.py

Question 10

Trouver tous les mots d'une chaîne de moins de 4 lettres

Code python :

```python
sentence = 'On a summer day somner smith went simming in the sun and his red skin stung'

examine = sentence.split()

result = [word for word in examine if len(word) <4]
print(result)

sentence = 'On a summer day somner smith went simming in the sun and his red skin stung'
words=tuple(sentence.split(' '))

result =[i for i in words if len(i) < 4]
print (result)

string = 'all the words in a string'
my_list = [word for word in string.split(' ') if len(word) < 4]
print(my_list)

```

q144.py

Question 11

Utilisez la compréhension d'une liste imbriquée pour trouver tous les nombres de 1 à 100 qui sont divisibles par n'importe quel chiffre à part 1 (2-9).

Code python :

```python
# old school
no_dups = set()
for n in range(1, 101):
    for x in range(2, 10):
        if n % x == 0:
            no_dups.add(n)
print(no_dups)
print()

# nested list comprehension

result = [
    number
    for number in range(1, 101)
    if True in [True for x in range(2, 10) if number % x == 0]
]
print(result)


numbers = list(range(1,101))
divisors = list(range(2,10))

ans = [n for n in numbers if any([ n % d == 0 for d in divisors])]
print(ans)

print([*set([i for i in range(1,101) for j in [2,3,4,5,6,7,8,9] if i % j == 0])])




#Just to be a smart-arse, using the full range 2-10 is unnecessary, because if a number isn't divisible by 2, it won't be divisible by 4, 6, 8 etc.
#So a simplified version of this is:

result = [n for n in range(1001) if 0 in [n % divisor for divisor in [2,3,5,7]]]

```

q145.py