Python Test Question Paper

aviral.np12@gmail.com Switch account

Draft saved

* Indicates required question

Email *

aviral.sharma@perficient.com

What is the output of print("Python"[-6])? * 1 point

- (●) P
- ( ) Y
- ( ) t
- ( ) Error

What is the correct way to concatenate two lists list1 and list2 in Python? * 1 point

- (●) list1 + list2
- ( ) list1.append(list2)
- ( ) list1.extend(list2)
- ( ) list1.concat(list2)

What is the output of print(3 == "3")? * 1 point

- ○ True
- ◉ False
- ○ Error
- ○ None

What is the output of print("Hello"[::-1])? * 1 point

- ○ "Hello"
- ◉ "olleH"
- ○ "lleH"
- ○ Error

import numpy as np *
arr = np.array([1, 2, 3, 4, 5])
condition = arr > 3
result = arr[condition]
print(result) 1 point

- ○ [1, 2, 3]
- ◉ [4, 5]
- ○ [True, True, True, False, False]
- ○ [False, False, False, True, True]

Write a Python script to read a text file data.txt, count the number of lines, words, and characters,space, count of each vowels greater than 3 in the file and then write these counts to another file output.txt * 5 points

```
def count_vowels(text):
    vowels = "aeiouAEIOU"
    vowel_counts = {}
    for vowel in vowels:
        vowel_counts[vowel] = 0

    for char in text:
        if char in vowels:
            vowel_counts[char] += 1

    result = {}
    for vowel, count in vowel_counts.items():
        if count > 3:
            result[vowel] = count

    return result

input_filename = "data.txt"
output_filename = "output.txt"

with open(input_filename, 'r') as file:
    data = file.read()

lines = 0
for char in data:
    if char == '\n':
        lines += 1
lines += 1

words = len(data.split())

characters = 0
for char in data:
    if char != '\n':
        characters += 1

spaces = 0
for char in data:
    if char == ' ':
        spaces += 1

vowel_counts = count_vowels(data)

with open(output_filename, 'w') as file:
    file.write("Number of lines: {}\n".format(lines))
    file.write("Number of words: {}\n".format(words))
    file.write("Number of characters: {}\n".format(characters))
    file.write("Number of spaces: {}\n".format(spaces))
    file.write("Vowels count greater than 3:\n")
```

```
    file.write("Vowel counts greater than 3:\n")
    for vowel, count in vowel_counts.items():
        file.write("{}: {}\n".format(vowel, count))

print("Counts have been written to {}".format(output_filename))
```

Write a Python code that takes 2 Lists and returns True if all elements in the list are unique, otherwise print a message accordingly. * 5 points

```
def check_same_elements(list1, list2):

    list1.sort()
    list2.sort()

    if len(list1) != len(list2):
        return False


    for i in range(len(list1)):
        if list1[i] != list2[i]:
            return False

    return True


list1 = input("Enter elements for list1: ").split()
list2 = input("Enter elements for list2: ").split()

# Convert input strings to integers (assuming elements are numeric)
list1 = [int(num) for num in list1]
list2 = [int(num) for num in list2]

# Call the function to check if lists have same elements
if check_same_elements(list1, list2):
    print("true")
else:
    print("List1 and list2 are not unique")
```

Create a DataFrame with missing values given below: * 5 points

'ProductID': [101, 102, 103, 104, 105],
'ProductName': ['Apple', 'Banana', 'Carrot', 'Doughnut', 'Eggplant'],
'Category': ['Fruit', 'Fruit', 'Vegetable', 'Bakery', None],
'Price': [1.2, None, 0.7, 1.5, 1.1],
'Stock': [30, 50, None, 15, 10]

a. Fill missing values with specified value and print
b. Sort the DataFrame by Price in descending order

```
import pandas as pd
import numpy as np

# Create the DataFrame with missing values
data = {
    'ProductID': [101, 102, 103, 104, 105],
    'ProductName': ['Apple', 'Banana', 'Carrot', 'Doughnut', 'Eggplant'],
    'Category': ['Fruit', 'Fruit', 'Vegetable', 'Bakery', None],
    'Price': [1.2, None, 0.7, 1.5, 1.1],
    'Stock': [30, 50, None, 15, 10]
}

df = pd.DataFrame(data)

# a. Fill missing values with specified value and print
filled_df = df.fillna({'Category': 'Unknown', 'Price': df['Price'].mean(), 'Stock':
df['Stock'].median()})
print("DataFrame with filled missing values:")
print(filled_df)
print()

# b. Sort the DataFrame by Price in descending order
sorted_df = df.sort_values(by='Price', ascending=False)
print("DataFrame sorted by Price in descending order:")
print(sorted_df)
```

Submit Clear form

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy

Google Forms