Assignment

August 22, 2021

0.1 1. List Remove Append

Description: Remove SPSS from input_list=[‘SAS’, ‘R’, ‘PYTHON’, ‘SPSS’] and add ‘SPARK’ in its place.

[1]:
```
## Solution 1:
input_list=['SAS', 'R', 'PYTHON', 'SPSS']
print('Before ' , input_list)
input_list.pop()
input_list.append('SPARK')
print('After ' , input_list)
```

```
Before  ['SAS', 'R', 'PYTHON', 'SPSS']
After  ['SAS', 'R', 'PYTHON', 'SPARK']
```

[2]:
```
## Solution 2:
input_list=['SAS', 'R', 'PYTHON', 'SPSS']
print('Before ' , input_list)
input_list[input_list.index('SPSS')]='SPARK'
print('After ' , input_list)
```

```
Before  ['SAS', 'R', 'PYTHON', 'SPSS']
After  ['SAS', 'R', 'PYTHON', 'SPARK']
```

0.2 2. String to List Conversion

Description: Convert a string input_str = ‘I love Data Science & Python’ to a list by splitting it on ‘&’. The sample output for this string will be:

[‘I love Data Science’, ’ Python’]

[3]:
```
input_str="I love Data Science & Python"
str_list=input_str.split("&")
print(str_list)
```

```
['I love Data Science ', ' Python']
```

0.3 3. Nested List

Description: Extract Python from a nested list

input_list = [[‘SAS’,‘R’],[‘Tableau’,‘SQL’],[‘Python’,‘Java’]]

[4]:
```
input_list = [['SAS','R'],['Tableau','SQL'],['Python','Java']]
print(input_list[2][0])
```

```
Python
```

0.4 4. It’s the time to disco

Description: t = (“disco”, 12, 4.5)

[5]:
```
t = ("disco", 12, 4.5)
print(t[0][2])
```

```
s
```

0.5 5. String Palindrome

Description: Write a program to check whether a string is a palindrome or not. Print 1 if

the string is a palindrome and 0 otherwise

0.5.1 Input: Any string

0.5.2 Output : 1 if the string is palindrome, 0 otherwise.

[6]:
```
def checkStringPalindrome(str_input):
        input_len=len(str_input)
        loop_control=int(input_len/2)
        for str_index in range(0, loop_control):
                if str_input[str_index] != str_input[input_len-str_index-1]:
                        return 0
        return 1

print(checkStringPalindrome('malayalam'))
print(checkStringPalindrome('abc'))
```

```
1
0
```

0.6 6. Reverse Words

Description: You will be given a sentence in the form of a string. You have to reverse the

order of the words in the sentence. Remember not to reverse the individual words, but

the order of words. Check the sample input-output for further clarification.

0.6.1 Input: A string, which will consist of a few spaces.

0.6.2 Output: The words in reverse order

[7]:
```
def reverseWordSentence(str_input):
        input_str_list=str_input.split(" ")
        loop_control=len(input_str_list)-1
        output_str_list=[]
        for str_index in range(loop_control,-1,-1):
                output_str_list.append(input_str_list[str_index])
        output_str = " ".join(output_str_list)
        return output_str

print(reverseWordSentence("I love programming"))
```

```
programming love I
```

0.7 7. String Formatting

Description: Write a program that satisfies below examples

0.7.1 Input 1: caloRie ConsuMed

0.7.2 Output 1: calorie_consumed

0.7.3 Input 2: data science

0.7.4 Output 2: data_science

0.7.5 Input 3: datascience

0.7.6 Output 3: datascience

[8]:
```
def stringFormatting(str_input):
        output_str=str_input.lower()
        output_str=output_str.replace(" ","_")
        return output_str

print(stringFormatting("caloRie ConsuMed"))
print(stringFormatting("data science"))
print(stringFormatting("datascience"))
```

```
calorie_consumed
data_science
datascience
```

1 Multiple Choice Questions

1.1 1. How will you extract 'love' from the string S = "I love Python"? (More than one option may be correct.).

1.1.1 a. S[2:5]

1.1.2 b. S[2:6]

1.1.3 c. S[3:7]

1.1.4 d. S[-11:-7]

1.1.5 e. S[-11:-8]

[9]:
```
## Answer 1. Option b,d
S = 'I love Python'
print(S[2:6])
print(S[-11:-7])
```

```
love
love
```

1.2 2. What will the output of 3 * 3 ** 3 be?

1.2.1 a. 9

1.2.2 b. 27

1.2.3 c. 81

1.2.4 d. 729

[10]:
```
## Answer 2. Option d. 81
print(3 * 3 ** 3 )
```

```
81
```

1.3 3. What will the output be of ((500//7) % 5) ** 3?

1.3.1 a. 1

1.3.2 b. 2.91

1.3.3 c. 71.42

1.3.4 d. 0

1.3.5 e. 8

[11]:
```
### Answer 3. Option a
print(((500//7) % 5) ** 3)
```

```
1
```

1.4 4. If you have a tuple T = (3, 5, 7, 11), what will the output of T.append(9) be?

1.4.1 a. (3, 5, 7, 9, 11)

1.4.2 b. (9, 3, 5, 7, 11)

1.4.3 c. (3, 5, 7, 11, 9)

1.4.4 d. Error

[12]:
```
## Answer 4. Option d. Error
T = (3, 5, 7, 11)
T.append(9)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-12-07118af80ae4> in <module>
      1 ## Answer 4. Option d. Error
      2 T = (3, 5, 7, 11)
----> 3 T.append(9)

AttributeError: 'tuple' object has no attribute 'append'
```

1.5 6. What will the output of the following code be?

l = [32, 34, 12, 27, 33]

l.append((l4, 19))

print(len(l))

1.5.1 a. 5

1.5.2 b. 6

1.5.3 c. 7

1.5.4 d. The code will throw an error

[13]:
```
## Answer 6. Option 6
l=[32, 34, 12, 27, 33]
l.append((14, 19))
print(len(l))
```

```
6
```

1.6 7. Which of the following statements is incorrect regarding sets in Python?

1.6.1 a. Sets do not contain duplicate elements

1.6.2 b. Sets are represented using curly braces

1.6.3 c. Sets are immutable

1.6.4 d. All of the above

1.7 Answer 7. Option a. Sets do not contain duplicate elements

1.8 8. What will be the output of following

D = {'l':['Raj', 22], '2':['Simran', 21], '3':['Rahul', 40]}

for val in D:

print(val)

1.8.1 a. 1

1.8.2 2

1.8.3 3

1.8.4 b. ['Raj', 22]

1.8.5 ['Simran', 21]

1.8.6 ['Rahul', 40]

1.8.7 c. 1 ['Raj', 22]

1.8.8 2 ['Simran', 21]

1.8.9 3 ['Rahul', 40]

1.8.10 d. 'Raj'

1.8.11 'Simran'

1.8.12 'Rahul'

[14]:
```
## Answer 8. Option a. 1\n2\n3
D={'l':['Raj', 22], '2':['Simran', 21], '3':['Rahul', 40]}
for val in D:
    print(val)
```

```
1
2
3
```

1.9 9. What will the 'comprehension equivalent' be for the following snippet of code?

for sentence in paragraph:

for word in sentence.split():

single_word_list.append(word)

1.9.1 a. word for sentence in paragraph for word in sentence.split()

1.9.2 b. [word for sentence in paragraph for word in sentence.split()]

1.9.3 c. word for word in sentence.split() for sentence in paragraph

1.9.4 d. [word for word in sentence.split() for sentence in paragraph]

1.10 Answer 9. Option b [word for sentence in paragraph for word in sentence.split()]

1.11 10. What will be the output of this code?

print(list(range(10,1,-1)))

1.11.1 a. [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

1.11.2 b. [9, 8, 7, 6, 5, 4, 3, 2]

1.11.3 c. [9, 8, 7, 6, 5, 4, 3, 2, 1]

1.11.4 d. [10, 9, 8, 7, 6, 5, 4, 3, 2]

[15]:
```
## Answer 10. Option d [10, 9, 8, 7, 6, 5, 4, 3, 2]
print(list(range(10,1,-1)))
```

```
[10, 9, 8, 7, 6, 5, 4, 3, 2]
```

[ ]:
```

```