6th Feb Assignment

February 10, 2023

1 Assignment 7

Q1. Create a function which will take a list as an argument and return the product of all the numbers after creating a flat list. Use the below-given list as an argument for your function.

- list1 = [1,2,3,4, [44,55,66, True], False, (34,56,78,89,34), {1,2,3,3,2,1}, {1:34, "key2": [55, 67, 78, 89], 4: (45, 22, 61, 34)}, [56, 'data science'], 'Machine Learning']
- Note: you must extract numeric keys and values of the dictionary also.

[3]:

```
list1 = [1,2,3,4, [44,55,66, True], False, (34,56,78,89,34), {1,2,3,3,2,1}, {1:
↪34, "key2": [55, 67, 78, 89], 4: (45,
22, 61, 34)}, [56, 'data science'], 'Machine Learning']
```

[41]:

```
def flatten(lis):
    flat_list=[]
    for i in lis:
        if type(i)==int or type(i)==float:
            flat_list.append(i)
        elif type(i)==tuple or type(i)==list or type(i)==set:
            flat_list.extend(flatten(i))
        elif type(i)==dict:
            for key,value in i.items():
                if type(key)==int or type(key)==float:
                    flat_list.append(key)
                if type(value)==int or type(value)==float:
                    flat_list.append(value)
                elif type(value)==list or type(value)==tuple:
                    flat_list.extend(flatten(value))
    return flat_list
```

[42]:

```
list2=flatten(list1)
```

[43]:

```
product=1
for i in list2:
    product=product*i
print(product)
```

```
4134711838987085478833841242112000
```

Q2. Write a python program for encrypting a message sent to you by your friend. The logic of encryption should be such that, for a the output should be z. For b, the output should be y. For c, the output should be x respectively. Also, the whitespace should be replaced with a dollar sign. Keep the punctuation marks unchanged.

Input Sentence: I want to become a Data Scientist.

Encrypt the above input sentence using the program you just created.

Note: Convert the given input sentence into lowercase before encrypting. The final output should be lowercase.

[51]:

```
def encryption(message):
    input_message=message.lower()
    encrypted_message=''
    for i  in input_message:
        if i.isspace()==1:
            encrypted_char='$'
        elif i.isalpha()==1:
            encrypted_char=chr(122-ord(i)+97)
        else:
            encrypted_char=i
        encrypted_message=encrypted_message+encrypted_char
    return encrypted_message
```

[52]:

```
input_sentence='I want to become a Data Scientist.'
```

[53]:

```
output=encryption(input_sentence)
```

[54]:

```
output
```

[54]:

```
'r$dzmg$gl$yvxlnv$z$wzgz$hxrvmgrhg.'
```