7th Feb Assignment

February 10, 2023

1 Assignment 8

Q1. You are writing code for a company. The requirement of the company is that you create a python function that will check whether the password entered by the user is correct or not. The function should take the password as input and return the string "Valid Password" if the entered password follows the below-given password guidelines else it should return "Invalid Password".

Note:

1. The Password should contain at least two uppercase letters and at least two lowercase letters.
2. The Password should contain at least a number and three special characters.
3. The length of the password should be 10 characters long.

[16]:

```
def check_password(password):
    uppercase_letters=0
    lowercase_letters=0
    number_count=0
    special_characters=0

    for i in password:
        if i.isupper():
            uppercase_letters+=1
        elif i.islower():
            lowercase_letters+=1
        elif i.isdigit():
            number_count+=1
        elif i in '!@#$%^&*':
            special_characters+=1

    if uppercase_letters>=2 and lowercase_letters>=2 and number_count>=1 and␣
 ↪special_characters>=3 and len(password)==10:
        print('Valid password')
    else:
        print('Invalid password')
```

[17]:

```
password='JaiprH8###'
```

```
[19]: check_password(password)
```

```
Valid password
```

Q2. Solve the below-given questions using at least one of the following:

1. Lambda function
2. Filter function
3. Zap function
4. List Comprehension

1.0.1 Check if the string starts with a particular letter

```
[22]: particular_letter='J'
      check_string=lambda x:print('Yes') if x[0]==particular_letter else print('No')
```

```
[25]: check_string('Jaiprakash')
```

```
Yes
```

```
[26]: check_string('jimmy')
```

```
No
```

1.0.2 Check if the string is numeric

```
[32]: check_string_numeric=lambda i:i.isdigit()
```

```
[34]: string='123456'
      check_string_numeric(string)
```

```
[34]: True
```

```
[35]: string1='1234r'
      check_string_numeric(string1)
```

```
[35]: False
```

Sort a list of tuples having fruit names and their quantity. [(“mango”,99),(“orange”,80), (“grapes”, 1000)]

```
[43]: list1=[("mango",99),("orange",80), ("grapes", 1000)]
```

```
[45]: sorted_list=sorted(list1,key=lambda x:x[1])
```

```
[46]: sorted_list
```

```
[46]: [('orange', 80), ('mango', 99), ('grapes', 1000)]
```

Find the squares of numbers from 1 to 10

[47]:
```
[i*i for i in range(1,11)]
```

[47]:
```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

Find the cube root of numbers from 1 to 10

[48]:
```
cube_root=lambda a:pow(a,1/3)
```

[50]:
```
numbers=[1,2,3,4,5,6,7,8,9,10]
```

[55]:
```
print(list(map(cube_root,numbers)))
```

```
[1.0, 1.2599210498948732, 1.4422495703074083, 1.5874010519681994,
1.7099759466766968, 1.8171205928321397, 1.912931182772389, 2.0,
2.080083823051904, 2.154434690031884]
```

Check if a given number is even

[56]:
```
check_number=lambda x:print('Even') if x%2==0 else print('Odd')
```

[57]:
```
check_number(6)
```

```
Even
```

[58]:
```
check_number(7)
```

```
Odd
```

Filter odd numbers from the given list. [1,2,3,4,5,6,7,8,9,10]

[59]:
```
list2=[1,2,3,4,5,6,7,8,9,10]
```

[60]:
```
odd=lambda x:x%2!=0
```

[63]:
```
print(list(filter(odd,list2)))
```

```
[1, 3, 5, 7, 9]
```

Sort a list of integers into positive and negative integers lists. [1,2,3,4,5,6,-1,-2,-3,-4,-5,0]

[65]:
```
list3=[1,2,3,4,5,6,-1,-2,-3,-4,-5,0]
```

[69]:
```
positive=lambda x:x>=0
negative=lambda x:x<0
```

[74]:
```
positive_list=sorted(list(filter(positive,list3)))
```

[75]:

```
positive_list
```

[75]:

```
[0, 1, 2, 3, 4, 5, 6]
```

[76]:

```
negative_list=sorted(list(filter(negative,list3)))
```

[77]:

```
negative_list
```

[77]:

```
[-5, -4, -3, -2, -1]
```