

practise-python-july

October 9, 2023

```
[3]: print("ganesh")
```

ganesh

```
[2]: #list
list = [1, "ganesh", 30000,1]
print(list)
print(list*3)  ## repats list 3 times
list1=[2,"ok"]
print(list[1:])  ## slicing
print(list[:-2])  ## negative slicing
print(len(list))  ## length of the list (list.length in scala)
list.append(6)  ## add and element at the last (5 :: list in scala)
print(list)
list.reverse()  ## reverse the list ( list.reverse in scala)
print(list)
print(list.count(1))  ## counts the occurence of element in the list
list1.extend(list)  ## extends the list by attaching other list ( concat, ::: ,
↳ ::: in scala)
print(list1)
list.remove("ganesh")  ## removes specific element from the list ( filter in
↳ scala)
print(list)
list.sort()  ## sorts the elemnts (list.sorted in scala)
print(list)
list.pop()  ## removes the last element
print(list)
print(list.index(6))  ## gives the index position of specific element (list.
↳ indexOf(3) in scala)
list.insert(2, "fury")  ## inserts the element at out choice index position
print(list)
```

```
[1, 'ganesh', 30000, 1]
[1, 'ganesh', 30000, 1, 1, 'ganesh', 30000, 1, 1, 'ganesh', 30000, 1]
['ganesh', 30000, 1]
[1, 'ganesh']
4
[1, 'ganesh', 30000, 1, 6]
```

```
[6, 1, 30000, 'ganesh', 1]
2
[2, 'ok', 6, 1, 30000, 'ganesh', 1]
[6, 1, 30000, 1]
[1, 1, 6, 30000]
[1, 1, 6]
2
[1, 1, 'fury', 6]
```

```
[31]: #remove duplicates from list
list1=[1,"ganesh",30000,2,1,"ganesh", "josh"]
print(list1, list1[2])
set1=set(list1)
print(set1)
```

```
[1, 'ganesh', 30000, 2, 1, 'ganesh', 'josh'] 30000
{1, 2, 'ganesh', 30000, 'josh'}
```

```
[40]: #duplicates n unq values in a list
list1=[1,"ganesh",30000,2,1,"ganesh", "josh"]
print(list1)
unq=[]
dup=[]
for i in list1:
    if i not in unq:
        unq.append(i)
    else :
        dup.append(i)
print(unq)
print(dup)
```

```
[1, 'ganesh', 30000, 2, 1, 'ganesh', 'josh']
[1, 'ganesh', 30000, 2, 'josh']
[1, 'ganesh']
```

```
[105]: #tuple
tuple1=(1,"ganesh",3000)
print(tuple1)
#tuple1[1]="fury"
print(tuple1[1:])
print(len(tuple1))
```

```
(1, 'ganesh', 3000)
('ganesh', 3000)
3
```

```
[30]: #dictionary
dict={"name":"ganesh", "age":23, "ofc":"tcs"}
```

```

print(dict["name"]) ##(map.get("name") in scala)
dict["id"]=1
print(dict)
dict["name"]="fury"
print(dict)
print(dict.keys()) ## gets dict keys (map.keys in scala)
print(dict.values()) ## gets dict values (map.values in scala)
print(len(dict)) ## gets dict length
print(dict.items()) ## gets all keys and values of dict
dict1=dict.copy() ## copies the dictionary to other dictionary
print(dict1)
dict.clear() ## cleares the dictionary
print(dict)

```

```

ganesh
{'name': 'ganesh', 'age': 23, 'ofc': 'tcs', 'id': 1}
{'name': 'fury', 'age': 23, 'ofc': 'tcs', 'id': 1}
dict_keys(['name', 'age', 'ofc', 'id'])
dict_values(['fury', 23, 'tcs', 1])
4
dict_items([('name', 'fury'), ('age', 23), ('ofc', 'tcs'), ('id', 1)])
{'name': 'fury', 'age': 23, 'ofc': 'tcs', 'id': 1}
{}

```

```

[115]: #string formatting
str="hello"
print(str)
print(str[1:]) ## (str.substring(1,6) in scala)
print(str + " " + "ganesh")
str="fury"
print(str)
str1="gani"
print(str1)
print(str+str1) ## (str1.concat(str2) in scala)
print(str*2)
print("i am %s age is %d" %("ganesh",23))
print(len(str))

```

```

hello
ello
hello ganesh
fury
gani
furygani
furyfury
i am ganesh age is 23
4

```

```
[96]: #string palindrome or not
str="arora"
print(str)
print(str[2])
print(str[::-1])
#str[1]="k"  ## string are immutable
if str==str[::-1]:
    print("palindrome")
else:
    print("not palindrome")

## (or)
str="ganesh"
str1=""
for i in str:
    str1=i+str1  ## str concating
print(str1)
```

```
arora
o
arora
palindrome
hsenag
```

```
[3]: #common elements between strings
str1="ganesh"
str2="furygani"
print(str1)
print(str2)
s1=set(str1)
s2=set(str2)
lst=s1 & s2
print(lst)
```

```
ganesh
furygani
{'n', 'g', 'a'}
```

```
[6]: #common elemnts btw strings
str1="ganesh"
str2="furygani"
print(str1)
print(str2)
for i in str1:
    if i in str2:
        print(i)
    else:
```

```
pass
```

```
ganesh  
furygani  
g  
a  
n
```

```
[19]: ##Convert list to string  
list=["fury","gani","ok"]  
sep=""  
str=sep.join(list)  
print(str)
```

```
furyganiok
```

```
[47]: #string functions  
str=" ganesh-2023-07-24 "  
print(str)  
str1=str.split("-")  ## string to array  
print(str1)  
print(list(str))      ## string to list  
str2="-".join(str1)  ## array or list to string  
print(str2)  
print(str.endswith("24 "))  
print(str.splitlines())  
print(len(str))  
print(max(str))  
print(str.replace("07", "08"))  
print(str.title())  
print(str.capitalize())  
print(str.strip())      ## removes leading and lagging spaces  
print(str.rstrip())  
print(str.find("07"))    ## return -1 if not found  
print(str.index("24",1)) ## raises exception if not found  
print(str.center(40,"."))  
print(str.count("2",1,len(str)))  
print(str.upper())  
print(str.isalnum())  
print(str.swapcase())  
print(str + str2)      ## concatting two strings
```

```
ganesh-2023-07-24  
[' ', 'g', 'a', 'n', 'e', 's', 'h', '-', '2', '0', '2', '3', '-', '0', '7', '-',  
'2', '4', ' ']  
[' ganesh', '2023', '07', '24 ']  
ganesh-2023-07-24  
True
```

```
[' ganesh-2023-07-24 ']
19
s
ganesh-2023-08-24
Ganesh-2023-07-24
ganesh-2023-07-24
ganesh-2023-07-24
ganesh-2023-07-24
13
16
... ganesh-2023-07-24 ...
3
GANESH-2023-07-24
False
GANESH-2023-07-24
ganesh-2023-07-24 ganesh-2023-07-24
```

```
[9]: ## regex
import re
str1="Fury Gani is a pro player pro and pro"
pattern="pro"
print(re.search(pattern , str1)) ## matches anywhere in the string
print(re.match(pattern , str1)) ## matches only at the start of the string
print(re.sub(pattern, "noob", str1)) ## replaces all or else we can give max_
↪value in last
```

```
<re.Match object; span=(15, 18), match='pro'>
None
Fury Gani is a noob player noob and noob
```

```
[16]: ## switch/case statement

x=4
match x:
    case 1: print("one")
    case 2: print("two")
    case 3: print("three")
    case _: print("invalid")
```

```
invalid
```

```
[26]: List=[1,2,3,4,5]
even_list=[]
odd_list=[]
for i in List:
    if i%2==0:
        even_list.append(i)
    else:
```

```

        odd_list.append(i)
print(even_list, odd_list)
## or
for i in List:
    if i%2==0:
        List.remove(i)
    else:
        pass
print(List)

## for comprehension
List1=[1,2,3,4,5]
x = [i for i in List1 if i%2==0] ## similar to yield we will get output as list
    ↪ here
print(x)

```

```

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

```

```

[33]: ## generators with yield

def generator(n):
    for i in range(1, n+1):
        yield i

for num in generator(5): ## finally in place of generator(5) a list with 1 to 5
    ↪ will be placed from yield
    print(num)

```

```

1
2
3
4
5

```

```

[52]: ## fibanocci using generators

```

```

def fib(n):
    a,b=0,1
    while(n>0):
        yield a
        a,b=b,a+b
        n-=1
for i in fib(10):
    print(i)

```

```

0

```

1
1
2
3
5
8
13
21
34

```
[41]: # decorators

def my_decorator(func):
    def wrapper():
        print("Hello ganesh")
        func()
        print("Ok Bye")
    return wrapper

@my_decorator
def say_hello():
    print("how are you")

say_hello()
```

Hello ganesh
how are you
Ok Bye

```
[84]: ## second target element in a string
str1="ganesh"
k=list(str1) ## string to list
k.sort()
print(k[-2])
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[84], line 3
      1 ## second target element in a string
      2 str1="ganesh"
----> 3 k=list(str1) ## string to list
      4 k.sort()
      5 print(k[-2])

TypeError: 'list' object is not callable
```



```
[58]: ## second target element in a list
a=[1,2,3,8,5,6]
def snd_largest(list):
    if len(list)<2:
        return None
    else:
        lrg, s_lrg=0,0
        for num in list:
            if num>lrg:
                s_lrg=lrg
                lrg=num
            elif num>s_lrg and num != lrg:
                s_lrg=num
        return s_lrg
snd_largest(a)
```

[58]: 6

```
[37]: ## Basic operations
## even or odd
Array=[1,2,3,4,5]
for i in Array:
    if i%2==0: print(f"{i} is even number.")
    else: print(f"{i} is odd number.")
## prime or not
for i in Array:
    def func(i):
        if i<=1:
            return False
        else:
            for j in range(2,i-1):
                if i%j==0:
                    return False
            return True
    res=func(i)
    print(f"{i} is prime:", res)

## perfect number or not
a=8
def perfect(n):
    if n<1:
        return False
    else:
        sum=0
        for i in range(1,n):
            if n%i==0:
                sum += i
```

```

        if sum==n:
            return True
        else:
            return False
res=perfect(a)
print(f"{a} is perfect:", res)

## factorial
b=6
def fact(m):
    if m<=1:
        return 1
    else:
        return m*fact(m-1)
print(fact(b))

## fibanocci

c=9
def fib(k):
    if k<=1:
        return k
    else:
        return fib(k-1)+fib(k-2)
print(fib(c))

## gcd

d=4
e=8
def gcd(a,b):
    if b==0:
        return a
    else:
        return gcd(b, a%b)
print(gcd(d,e))

```

```

1 is odd number.
2 is even number.
3 is odd number.
4 is even number.
5 is odd number.
1 is prime: False
2 is prime: True
3 is prime: True
4 is prime: False
5 is prime: True

```

```
8 is perfect: False
720
34
4
```

```
[2]: with open("file", "r+") as fo:
      data=fo.read(10)
      print(data)
      fo.close()
      with open("file", "w+") as fo1:
          fo1.write("bye bye")
          fo1.close()
      fo2=open("file", "a+")
      fo2.write("bye")
      fo2.close()
```

```
-----
FileNotFoundError                                Traceback (most recent call last)
Cell In[2], line 1
----> 1 with open("file", "r+") as fo:
      2     data=fo.read(10)
      3     print(data)

File
  ~\AppData\Roaming\Python\Python311\site-packages\IPython\core\interactiveshell..
  py:284, in _modified_open(file, *args, **kwargs)
    277 if file in {0, 1, 2}:
    278     raise ValueError(
    279         f"IPython won't let you open fd={file} by default "
    280         "as it is likely to crash IPython. If you know what you are
    doing, "
    281         "you can use builtins' open."
    282     )
--> 284 return io_open(file, *args, **kwargs)

FileNotFoundError: [Errno 2] No such file or directory: 'file'
```

```
[50]: ##Error handling
str=input("enter input:")
print(str)
a=10
b=0
try:
    a/b
except ArithmeticError:
    print("arithmetic error")
finally:
```

```

    print("Bye Bye")
## assertion ( santiy cehck) ( raise if false)
i=2
def func(i):
    assert (i>0),"no error"
    return i*2
func(i)

```

enter input:10
 10
 arthimetic error
 Bye Bye

[50]: 4

```

[61]: ##Basic operations
from collections import Counter
str1="silent"
str2="listen"
if Counter(str1)==Counter(str2):
    print("anagrams")
else:
    print("not anagrams")

```

anagrams

```

[107]: ## linkedin Anil varma G Questions
#1. square of each element in a list
a=[1,2,6,1,2,4]
def square(list):
    for j in list:
        yield j*j
for i in square(a):
    print(i)
## (or)
sqr_list=[]
for i in a:
    sqr_list.append(i*i)
print(sqr_list)

#2. count no of vowels in a string
str="ganesh"
count=0
lst=[]
for char in str:
    if char in "aeiouAEIOU":

```

```

        count+=1
        lst.append(char)
print(count)
print(lst)
print("".join(lst)) ## list to string

#3. count no of words from a given string
str1="ganesh is a good boy"
lst=str1.split()
print(len(lst))

'''str2="ganesh"
print(list(str2))
print(len(list1))'''

#4. count all +ve numbers from the list
list3=[-1, -2, 0, 3, 5]
List4=[]
count=0
for i in list3:
    if i>0:
        count+=1
        List4.append(i)
    else:
        pass
print(List4, count)

#5. create list of all first char from words in a string
str5="ganesh is a good boy"
lst=str5.split()
for i in lst:
    print(i[0])

## (or)
k=[word[0] for word in str5.split()]
print(k)

#6. removing vowels from string
str8="ganesh"
for i in str8:
    if i in "AEIOUaeiou":
        str8=str8.replace(i, "")
print(str8)

#7. frequency of each word in a string
str9="abcbabsceh"
dct={}

```

```

for i in str9:
    if i in dct:
        dct[i]+=1
    else:
        dct[i]=1
print(dct)

## (or)
for i in str9:
    print(i,str9.count(i))

#8. removing punctaions from staring
str1="ga@#$nesh"
punc="!@#$%^&*"
str2=""
for i in str1:
    if i not in punc:
        str2=str2+i
print(str2)

```

```

1
4
36
1
4
16
[1, 4, 36, 1, 4, 16]
2
['a', 'e']
ae
5
[3, 5] 2
g
i
a
g
b
['g', 'i', 'a', 'g', 'b']
gnsh
{'a': 2, 'b': 3, 'c': 2, 's': 1, 'e': 1, 'h': 1}
a 2
b 3
c 2
b 3
a 2
b 3

```

s 1
c 2
e 1
h 1
ganesh