Functional Programming

Monday, July 28, 2025 7:26 PM

It is used to convert the simple instruction codes into more shorter form.

1. Lambda
2. Map
3. Filter

1. Lambda: It is an anonymous function that is used to perform simple operations in python.

   Syntax:

   i. Var= lambda args: expression
   or
   ii. Var= lambda args: TSB if Condn else FSB

   Eg:

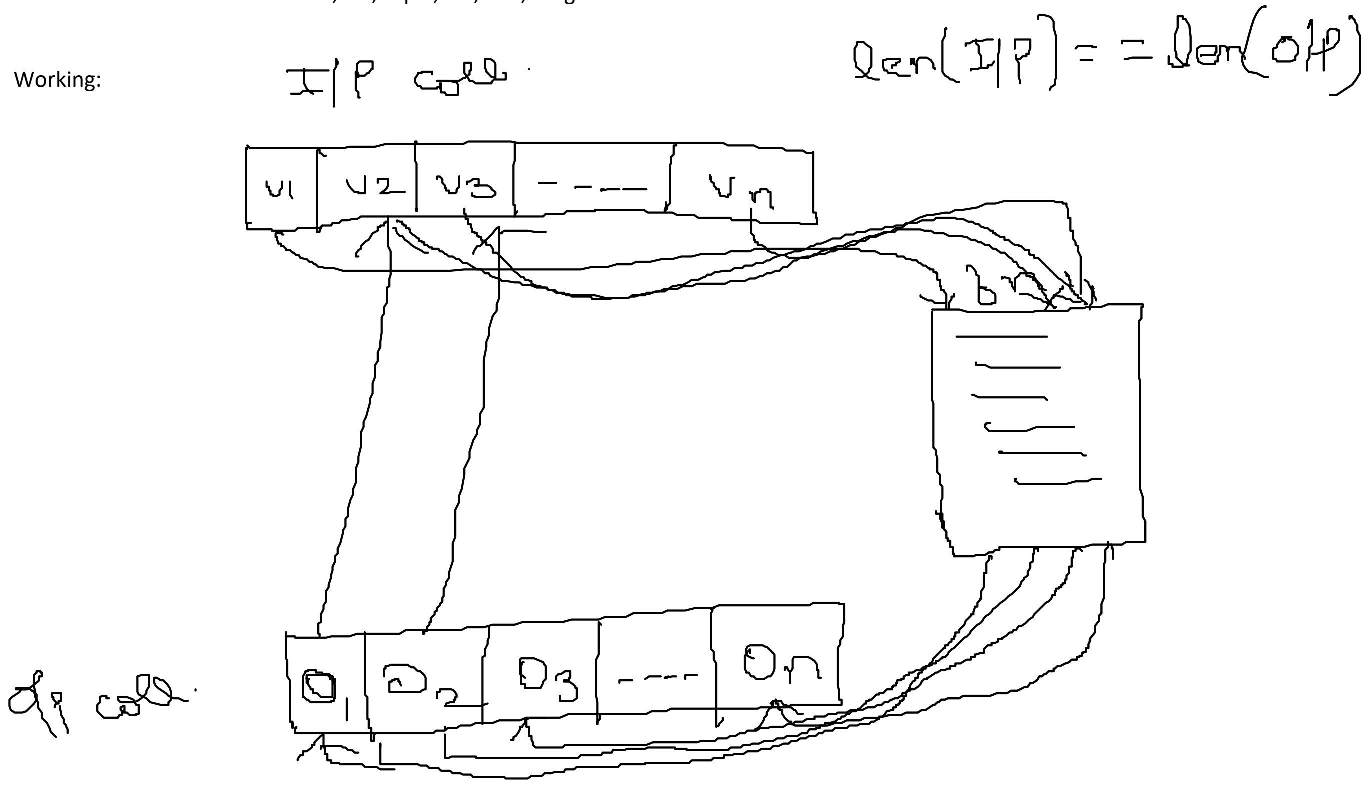
```
#Even
'''even=lambda n:n%2==0
print(even(10))'''

#Even or odd
'''no=lambda n:'Even' if n%2==0 else 'Odd'
print(no(int(input('Enter a number: '))))'''

#add 2 numbers
'''add= lambda a,b:a+b
print(add(int(input('Enter the number: ')),int(input('Enter the number: '))))'''

#WAP to check whether the string is starting from vowel
'''check= lambda s:s[0] in 'aeiouAEIOU'
print(check('Apple'))'''

#WAP to check whether the value is in list or not
'''val=lambda v,l: v in l
print(val(10,[10,20,30]))'''

#Q. WAP to return n+10 if n is multiple of 5 else return n-10
'''mul=lambda n: n+10 if n%5==0 else n-10
print(mul(20))'''

#Q. WAP to check whether the string is palindrome or not, if it is print the
#string if not print reverse of the string.
'''pal= lambda s:s if s==s[::-1] else s[::-1]
print(pal('namana'))'''

#Q. WAP to add minimum 2 and maximum 5 numbers.
'''add= lambda a,b,c=0,d=0,e=0: a+b+c+d+e
print(add(2,3,4,5,6))'''

#Q. WAP to return the concatinated string if the length of both the strings
#are same else print the first string.
ret=lambda s1,s2: s1+s2 if len(s1)==len(s2) else s1
print(ret('hi','bye'))
```

2. Map(): If we want to work on collection data then we use map function. It works similar to looping statements.

   Syntax:
   Var= map(fname, collection)

- In fname we can use inbuilt, user-defined, lambda function.
- In collection we can use str, list, tuple, set, dict, range

Working:



Variable gives address, to get the value we need typecasting.

Eg:

```
#WAP to find the square of the nos. from 1 to 10
'''sqr= lambda n:n**2
sqr_val= map(sqr, range(1,11))
print(list(sqr_val))'''

'''sqr_val= map(lambda n:n**2, range(1,11))
print(list(sqr_val))'''

'''print(list(map(lambda n:n**2, range(1,11))))'''


#WAP to get the following output
#I/p l=['abcd','start','end','data']
#O/p [4,5,3,4]

#l=['abcd','start','end','data']
'''res= map(len, eval(input('Enter the list: ')))
print(list(res))'''

#WAP to find the factorial of all integers present in tuple
'''def fact(n):
   if n==1 or n==0:
      return 1
   return n*fact(n-1)
print(tuple(map(fact, eval(input('Enter the tuple: ')))))'''

#WAP to get the following output
#s='hii hello how'
#op= ['hi','ho','hw']

'''s='hii hello how'
```

s1=s.split()
sam= lambda s:s[0]+s[-1]
print(list(map(sam, s1)))'''

#WAP to get the following output
#{1:1,2:8,3:27,4:64,5:125}
'''cube=lambda n:(n,n**3)
print(dict(map(cube, range(1,6))))'''

3. Filter(): It also works on loops only but works on required data only and not on all the data.

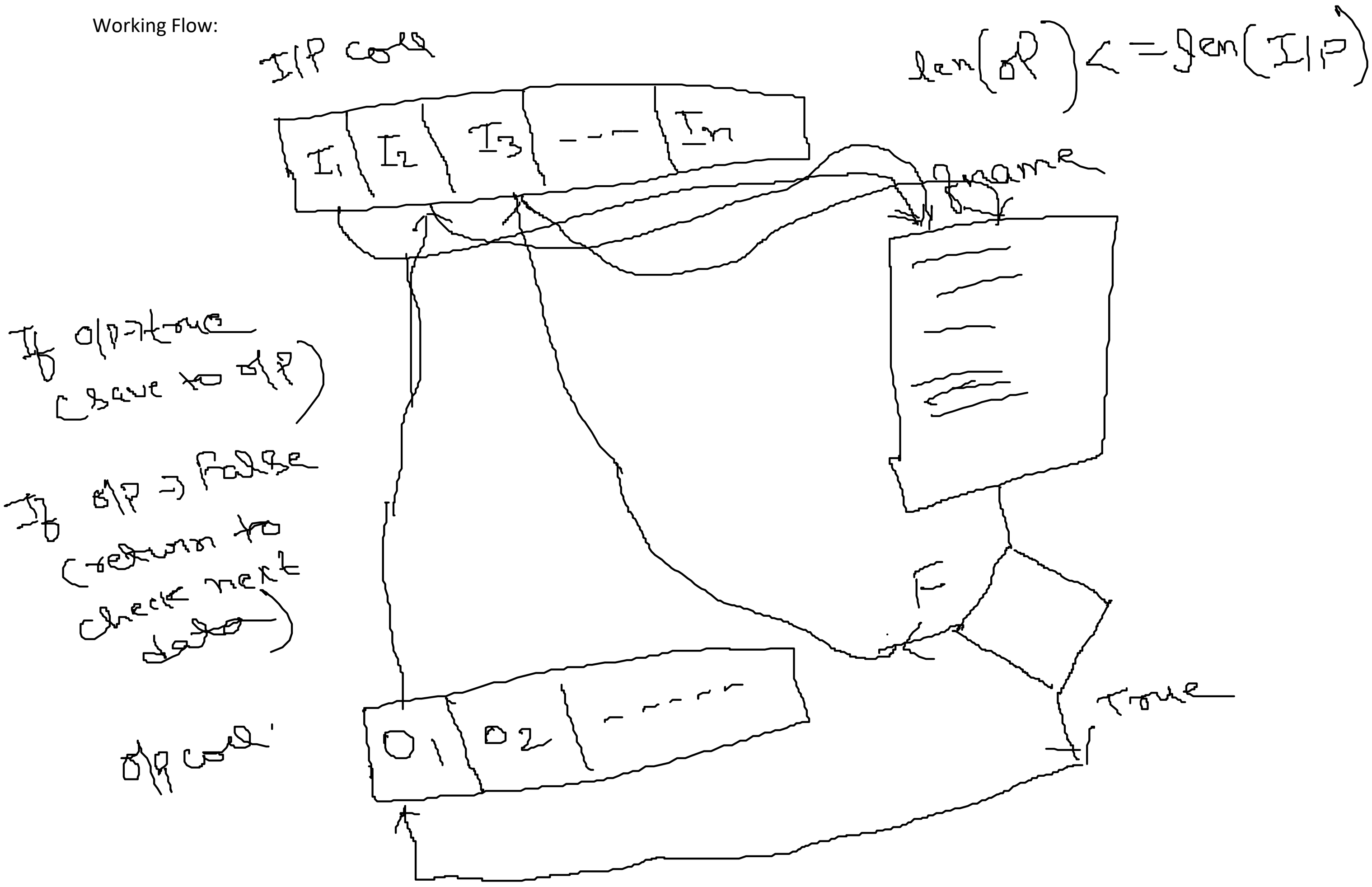
   - Here we can provide homogeneous as well as heterogeneous data.

   It is a function which is used to extract the required value from the collection.

   Syntax:
   Var= filter(fname, collection)
   Print(datatype(var))

   Here it gives the output in form of true and false internally.

   Working Flow:


I/P coll
len(O/P) <= len(I/P)
I1
I2
I3

In
fname
If o/p→true (save to o/p)
If o/p → False (return to check next data)
F
True
O1
O2

O/p coll

Eg:
#program to extract even numbers from 1 to 10
'''even=lambda n:n%2==0
even_n= filter(even, range(1,11))
print(list(even_n))'''

#WAP to extract all the string data item from tuple only if it is starting

```
#with lower lower case and ending with uppercase
'''sam=lambda v: type(v)==str and 'a'<=v[0]<='z' and 'A'<=v[-1]<='Z'
print(tuple(filter(sam, eval(input('Enter the tuple: ')))))'''

#WAP to extract all the collection values present in the list if they have even
#length
'''sam= lambda v:type(v) not in [int,float,complex,bool] and len(v)%2==0
print(list(filter(sam, eval(input('Enter the list: ')))))'''
```

Assignment:

1. WAP to extract all the prime numbers between 2 to 100.
2. WAP to find the square of all the even numbers present in the list.