**21/04/2019 Lecture29**

**Regular Expressions**

**#WAP to accept filename from user and print all words starting with t or T and ending with e or E**

#WAP to accept filename from user and print all words starting with t or T and ending with e or E

import re

def matchAlphabate(file\_name):

fd=open(file\_name)

data = fd.read()

regex\_obj=re.compile(r"\bT\w+e\b",re.IGNORECASE)

for match in regex\_obj.finditer(data):

print(match.start(),match.end())

def main():

file\_name = input("Enter File name");

matchAlphabate(file\_name)

if \_\_name\_\_=='\_\_main\_\_':

main()

**#WAP to accept filename from user and print all verbs in it** (\w+ing\b)

**#WAP to accept filename from user and print all words having ‘end’ in between.** e.g. Jeet**end**ra

**#WAP to** **accept a filename from user and print all digits from it.**

import re

def checkDigits(file\_name):

fd=open(file\_name)

data = fd.read()

regex\_obj=re.compile(r"\d+")

for match in regex\_obj.finditer(data):

print(match.start(),match.end())

def main():

file\_name = input("Enter File name");

checkDigits(file\_name)

if \_\_name\_\_=='\_\_main\_\_':

main()

**#WAP to** **accept a filename from user and delete all comments from it.**

(read file in buffer)

(start with # stop with \n use flag re.multiline)

(for multiline comment check starting with ‘’’/””””””)

(open same file in write mode and dump data)

#a or b => a|b

#a followed by one or more a or b => (a[ab]+)

#a followed by one or more a or b,not greedy => (a[ab]+?)

sequences without -, ., or space = > ([!-,.,])

sequences of lower case letter [a-z]+

sequences of upper case letter [A-Z]+

one upper case letter followed by lower case letters [A-Z+][a-z+]

a followed by anything, ending in b

**#WAP to accept file from user and print all words starting with capital letters**

**Lambda**

>>> x=lambda y: y\*y

>>> print(x(3))

9

>>> print(type(x))

<type 'function'>

>>> x=lambda y,z:[y\*y, z\*z]

>>> print(x(3,4))

[9, 16]

>>>

**map**

def square(x):

return x\*x

x = map(square,range(1,20,2))

print("map :")

for y in x:

print y

x=map(lambda x:x\*x,range(1,20,2))

print("map using lambda:")

for z in x:

print z

**FILTER**

def isEven(x):

return (x&1)==0

def square(x):

return x\*x

x=map(isEven,range(1,30))

print x

y=filter(isEven,range(1,30,2))

print y

z=filter(square,range(1,30))

print z

**REDUCE**

import functools

def multiply(x,y):

return x\*y

print functools.reduce(multiply,range(1,30))

**YIELD**

def YieldDemo():

i=0

for i in range(4):

yield i

x=YieldDemo()

print (next(x))

print (next(x))

print (next(x))

print (next(x))

print (next(x))

print (next(x))

#WAP **to accept a list of numbers from user and write a generator to return prime numbers from given list.**

**#Read presentation 10 AND practice given programs**