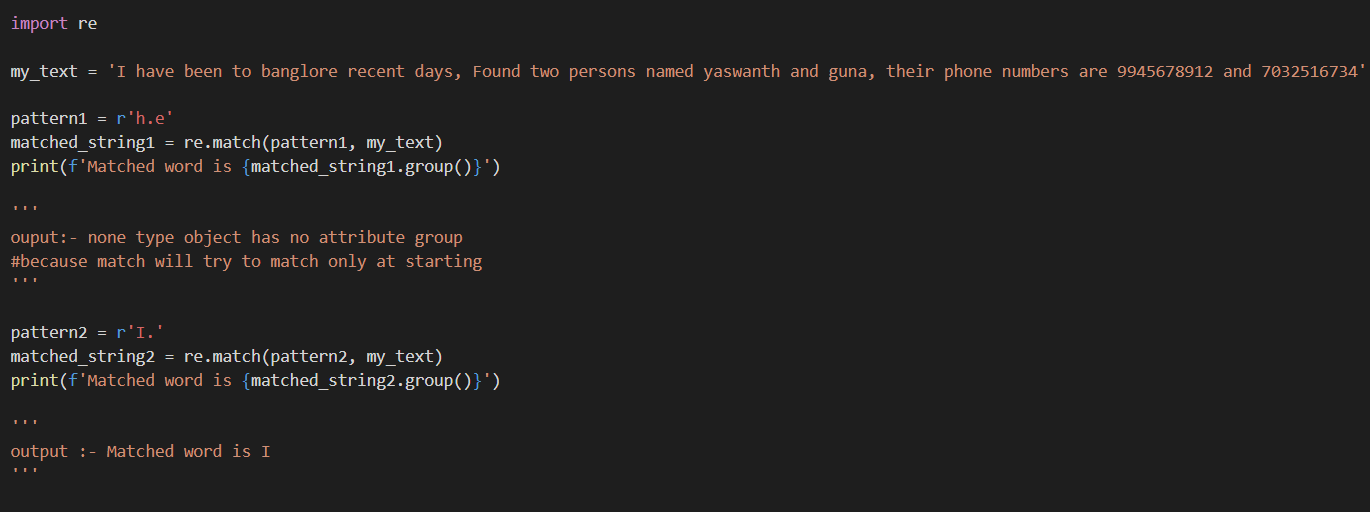
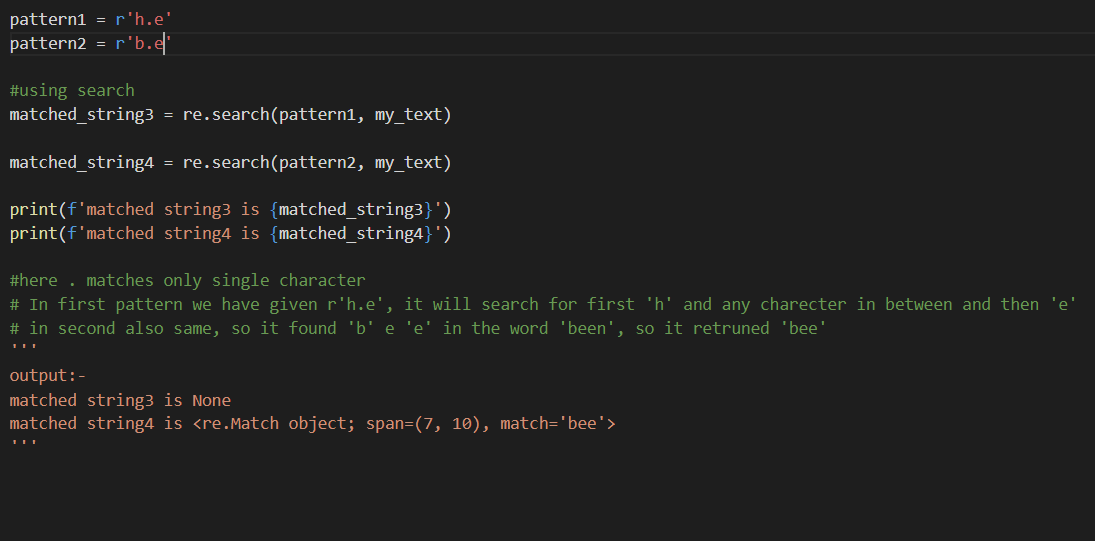
**#ReGex**

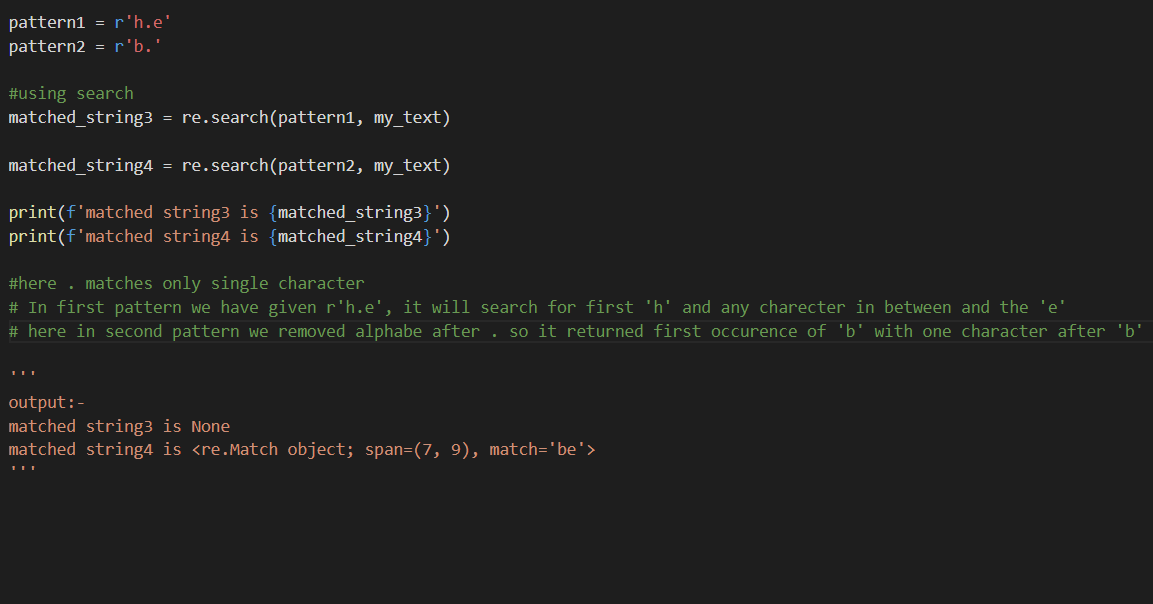
**#Using match**

****

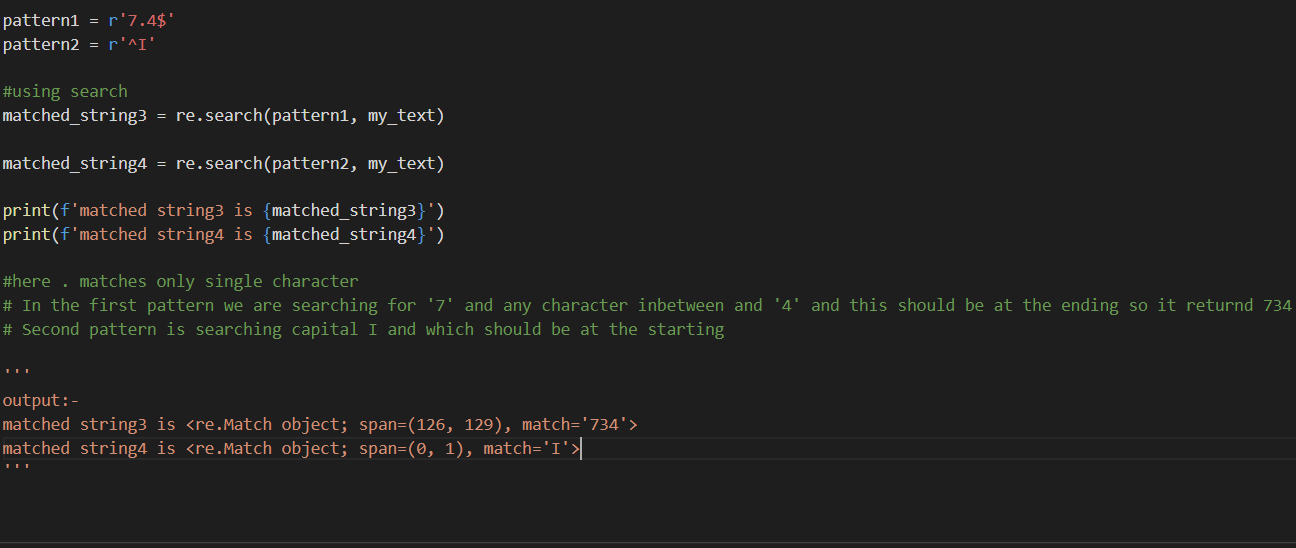
**# using search**

****

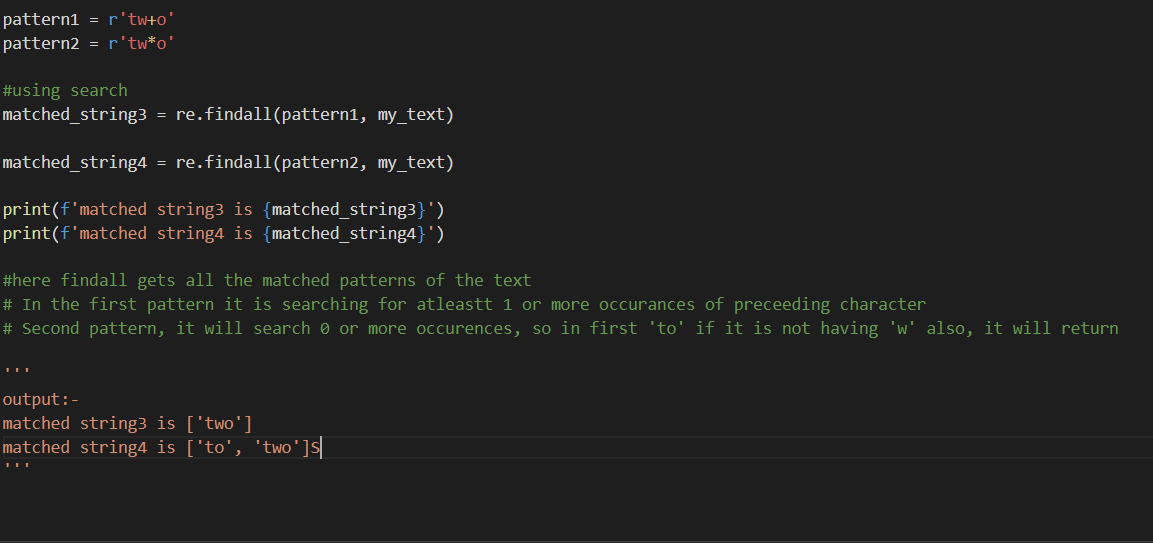
**#**

****

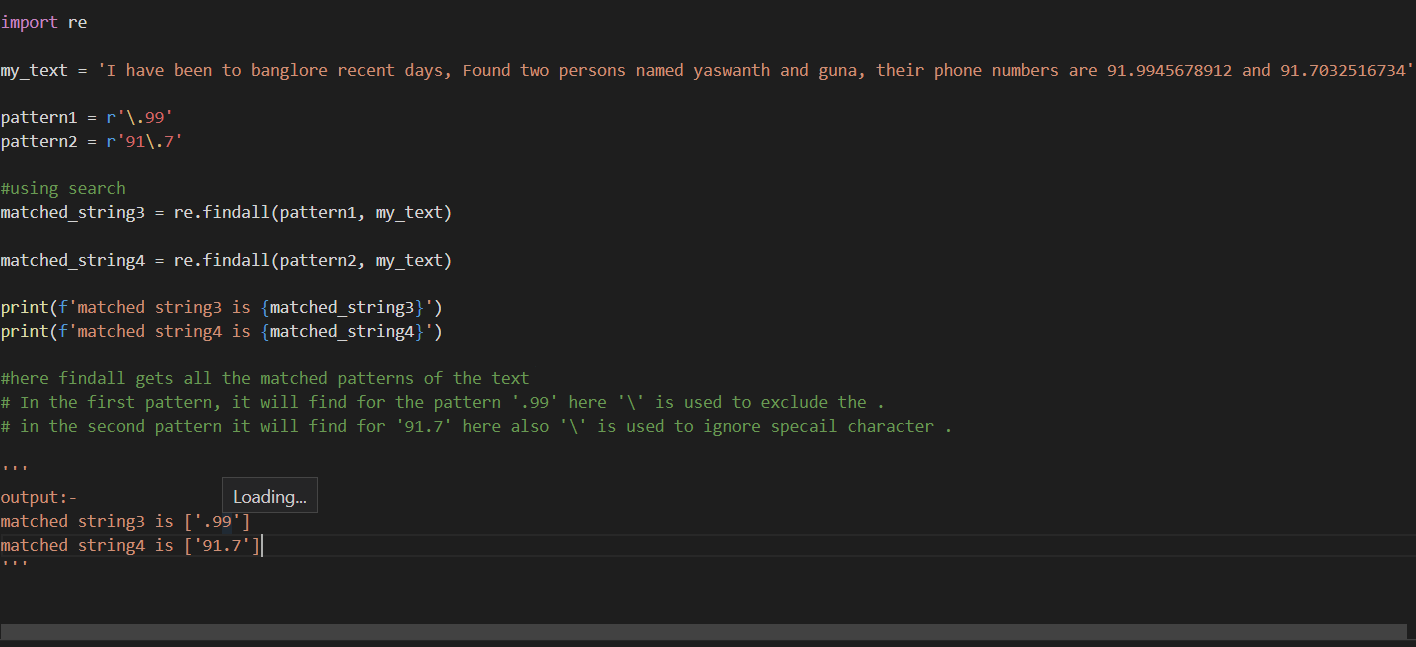
**# using ^ and $**

****

**#using +, \***

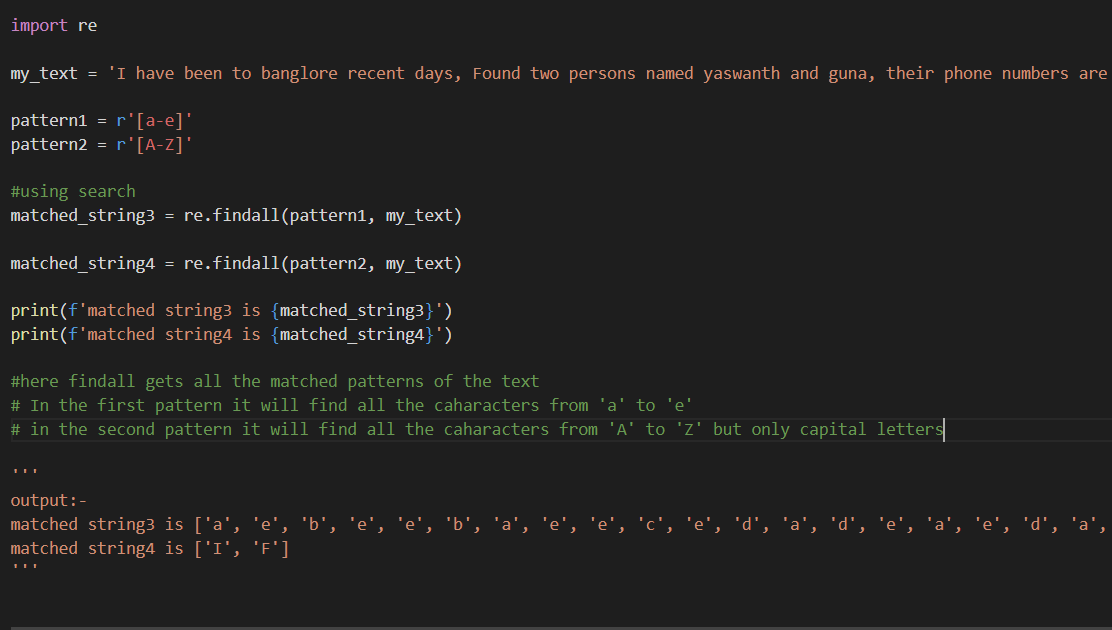
****

**#using \**

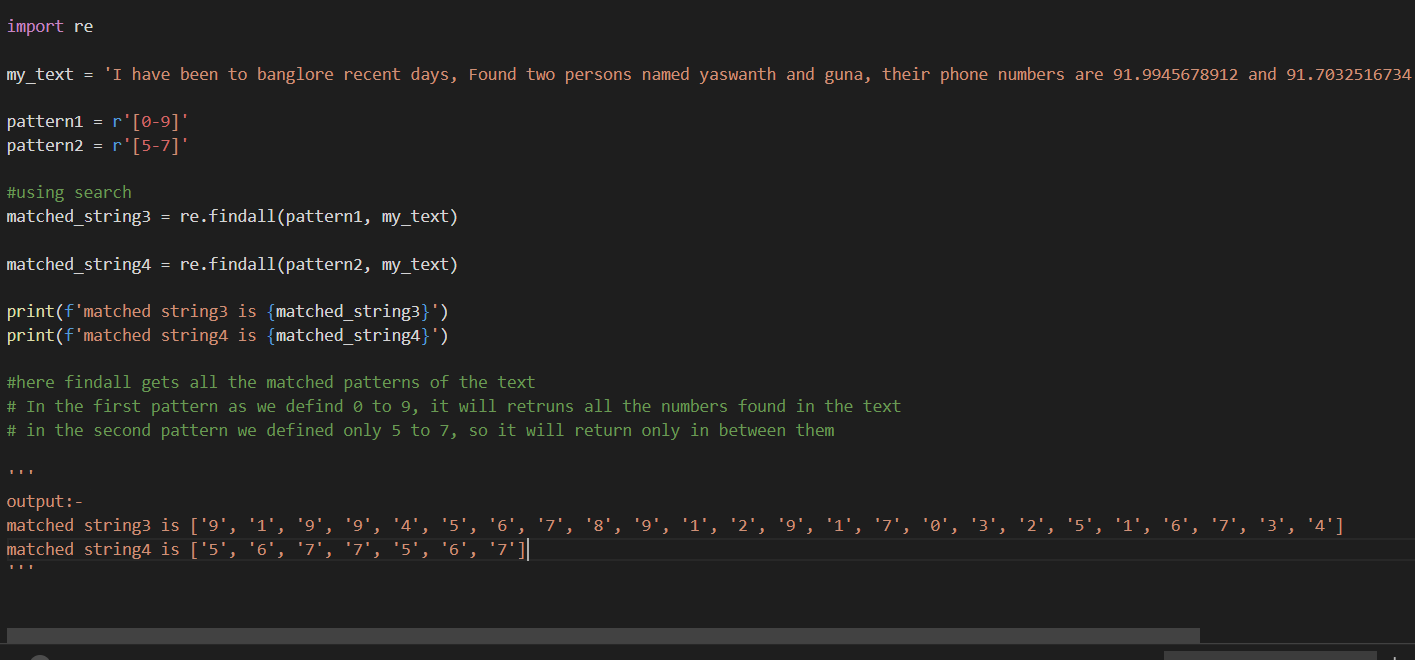
****

**#using characters**

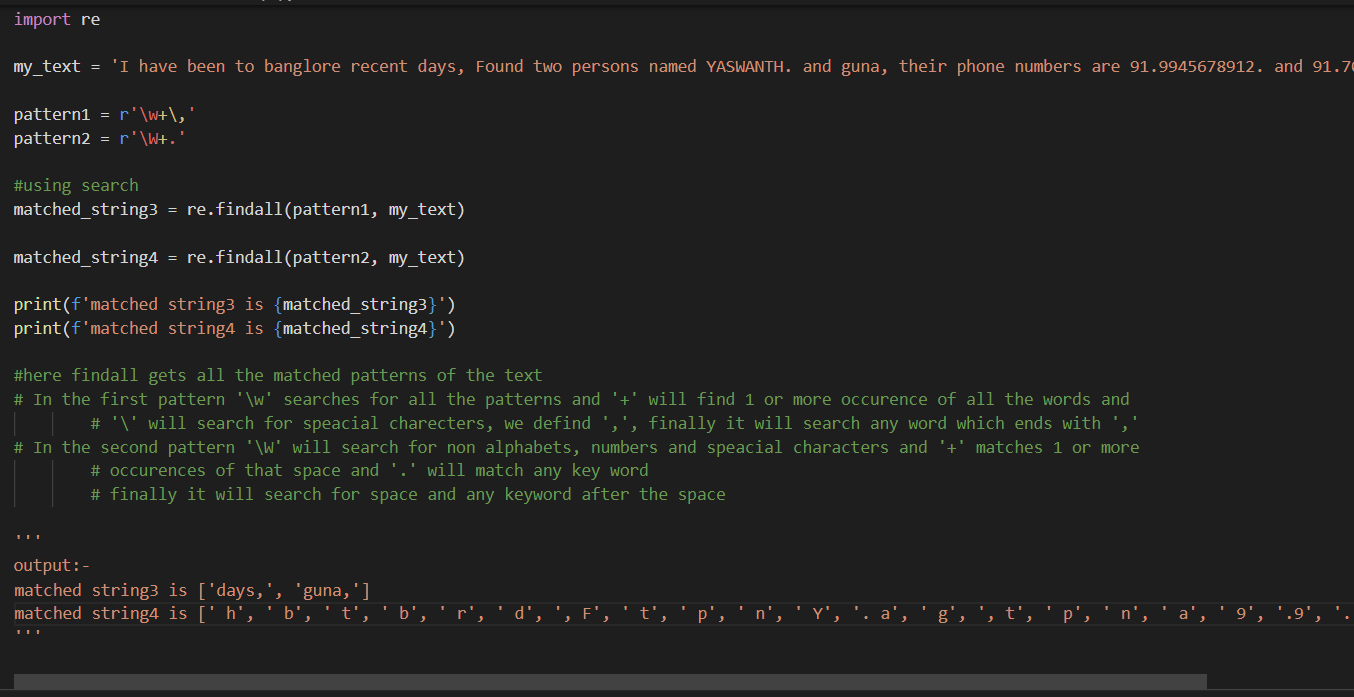
**#using [a-z] and [A-Z]**

****

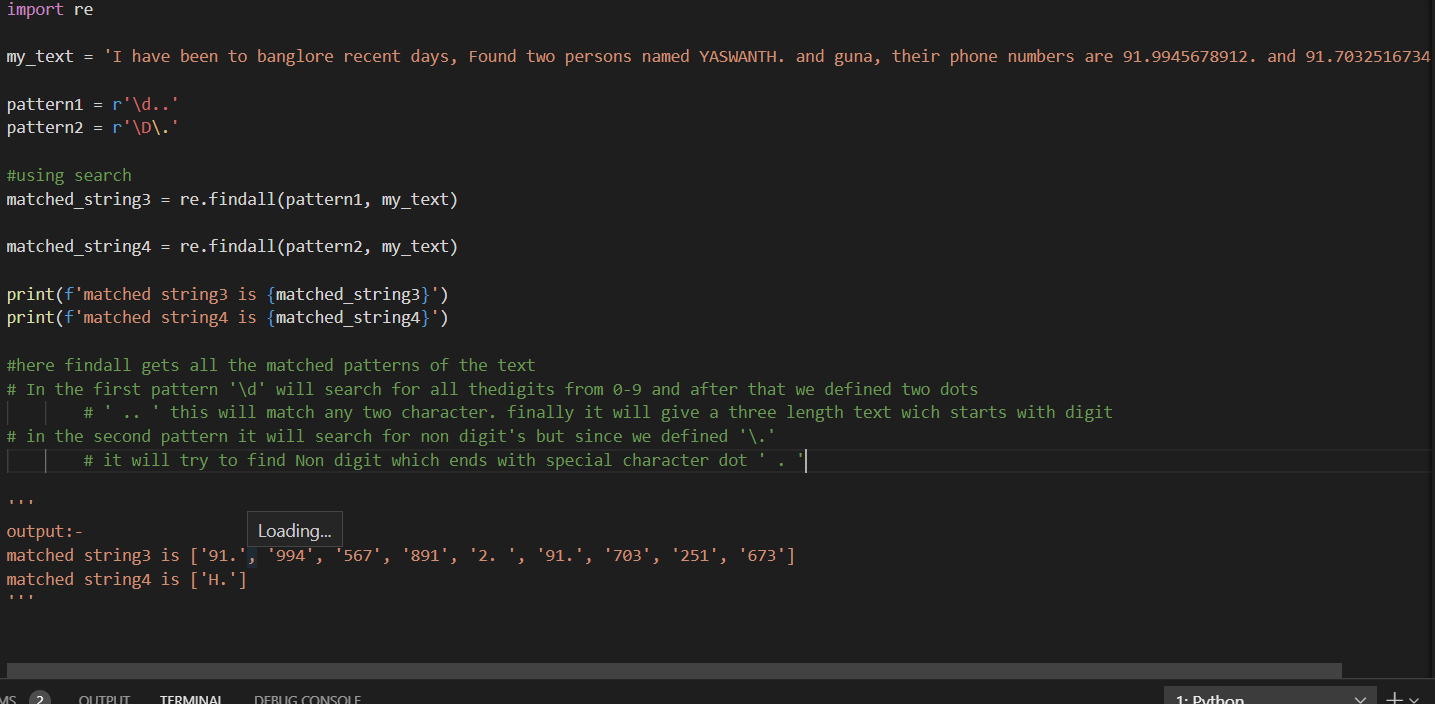
**#using [0-9] and [5-7]**

****

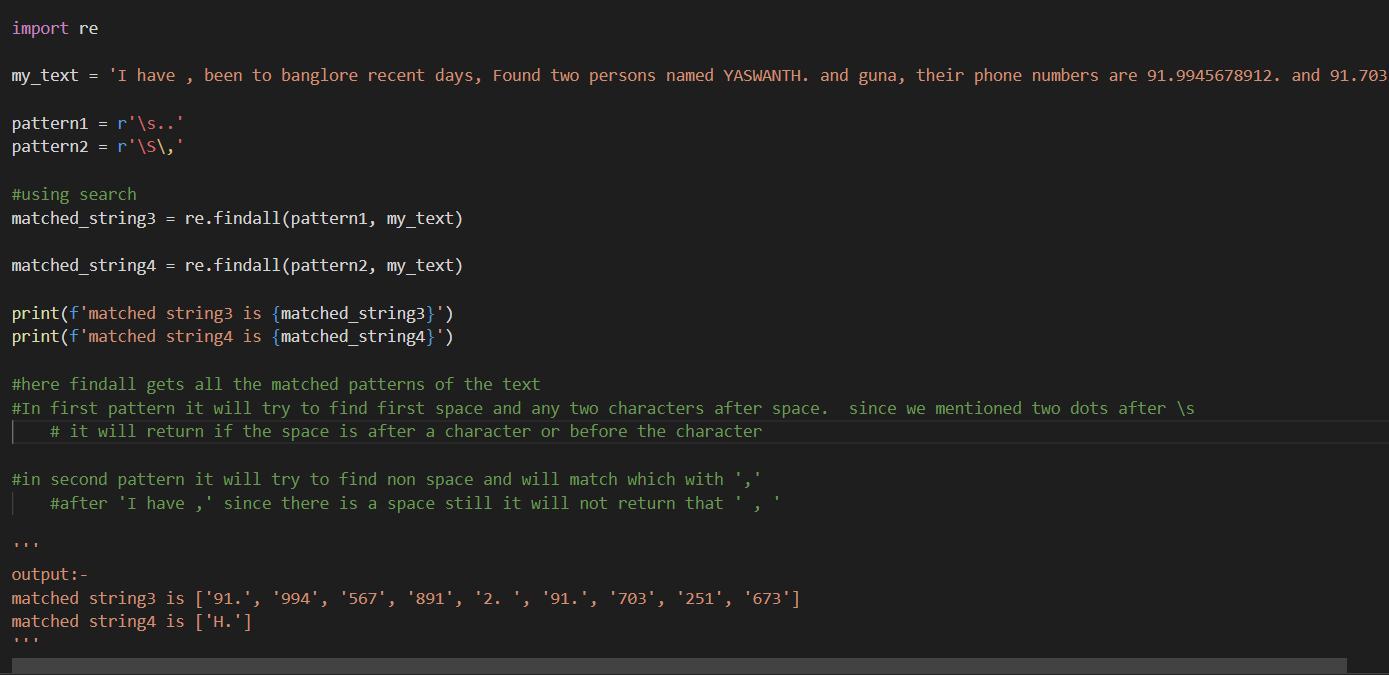
**#using \w and \W**

****

**#using \d and \D**

****

**# using \s and \S**

****

import logging

logging.basicConfig(filename = "c:/Users/w125682/Downloads/practice.log", level = logging.INFO,

format = '%(asctime)s - %(levelname)s - %(message)s', filemode = 'a')

logging.info('I am just an information log')

logging.debug('debugging log msg')

logging.warning('warning log msg')

logging.error('Error log msg')

logging.critical('Critical log msg')

import re

'''

my\_text = 'I have , been to banglore recent days, Found two persons named YASWANTH. and guna, their phone numbers are 91.9945678912. and 91.7032516734'

pattern1 = r'\s..'

pattern2 = r'\S\,'

#using search

matched\_string3 = re.findall(pattern1, my\_text)

matched\_string4 = re.findall(pattern2, my\_text)

print(f'matched string3 is {matched\_string3}')

print(f'matched string4 is {matched\_string4}')

#here findall gets all the matched patterns of the text

#In first pattern it will try to find first space and any two characters after space.  since we mentioned two dots after \s

    # it will return if the space is after a character or before the character

#in second pattern it will try to find non space and will match which with ','

    #after 'I have ,' since there is a space still it will not return that ' , '

output:-

matched string3 is ['91.', '994', '567', '891', '2. ', '91.', '703', '251', '673']

matched string4 is ['H.']

'''

def return\_str(data):

    logging.info(f"Given input is {data}")

    try:

        result = [item for  item in data if type(item)==str]

        return result

    except Exception as err:

        logging.error(f"Error is {err}")

result = return\_str(['mahesh', 1, 2, 'yash'])

#logger.info(f"Strings in the given list is {result}")

#print(f"retunred str is {return\_str(['mahesh', 1, 2, 'yash'])}")

'''

output :- retunred str is ['mahesh', 'yash']

'''

def return\_int(data):

    logging.info(f"given data is {data}")

    try:

        #result = [item for  item in data if type(item)==int]

        pattern = r'\d+'

        result = re.findall(pattern, str(data))

        print(result)

        result = [int(item) for item in result]

        #logging.info(f"Intiger's in the given data is {result}")

        return result

    except Exception as err:

        logging.error(f"Error while getting intiger's in the given list {err}")

print(f"retunred str is {return\_int(['mahesh', 1, 2, 'yash', (12, 34, 45), {67, 45, 89}, {'mahesh':23}])}")

"""

output :- retunred str is ['1,', '2,', '12', '34', '45', '89', '67', '45', '23']

"""