Day Objectives

- Regular Expressions
    - Constructing Regular Expressions for Various use cases
    - Regular Expressions Module and related in python
- Improving the Contacts Application with name and phone number validation
- File Handling
    - Text Files
    - Upgrading the Contacts Application to store contact information in a text file

Regular Expressions

- Pattern Matching
- Symbolic Notation of a pattern
    - Pattern : Format which repeates
    - Pattern(RE) - The set of all strings natches that pattern
- [0-9] -> Any Digit
- [a-z] -> Any lower case alphabet
- [2468]-> All single multiples of 2

- ^[0-9]{1}$ -> only single digit numbers hilighting
- ^[0-9]{3}$ -> only 3 digit numbers hilighting
- ^[0-9]{4}$ -> only 4 digit numbers hilighting
- [0-9]*0$ -> All multiples of 10

- Task

============

- `^([1-9][0-9]*[05])|([5])$` -> All multiples of 5
- `^[0-9]{10}$` ->All 10 digit numbers
- [w][o][r][d] or (word) -> Searching for a word
- `^[6-9][0-9]{9}$|^[0][6-9][0-9]{9}|^[+][9][1][6-9][0-9]{9}$` validating phone number(India)(start with 9876
- Email Validation (username@domain.com)
    - Username(`^[0-9a-z][0-9a-z_.]{4,13}[0-9][a-z]$`)
        - Length of username : [6,15]
        - No Special Characters other than _ .
        - Should not begin and end with _ .
        - Character set : All digits and lower case alphabet _ .
    - domain
        - Length of domain : [3,18]
        - No Special characters
        - Character set : All digits and lower case alphabet
    - extension
        - Length of extension: [2,4]
        - No Special characters
        - Character set : lower case alphabet

`^[a-zA-z0-9._]+@[A-Za-z].+[a-z]$`

--> ^[a]...[z]$ ANy string of length 5 that starts with 'a' and ends with

In [ ]:

```
```

In [9]:

```
# Function to validate a phone number in python
import re
def phoneNumberValidator(number):
    pattern = '^[6-9][0-9]{9}$|^[0][6-9][0-9]{9}|^[+][9][1][6-9][0-9]{9}$'
    if re.match(pattern,str(number)):
        return True
    return False
phoneNumberValidator(8500782761)
```

Out[9]: True

In [29]:

```
contacts = {}
def addContact(phone,email):
    # Verify that contact doesnot already exist
    li=[]
    li.append(phone)
    li.append(email)
    return li

#     if name not in contacts and phoneNumberValidator(phone):
#         contacts[name] = li
#         print("contact %s added" % name)
#     if name in contacts:
#         print("Contact %s already exists" % name)
#     elif not phoneNumberValidator(phone):
#         print("Phone number is inValid")
# addContact("vali","9502304797")
li=addContact(9989794454,"mastan@gmail.com")
li[0]
```

Out[29]: 9989794454

In [32]:

```
def addDetails(name,li):
    if name not in contacts and phoneNumberValidator(li[0]):
        contacts[name] = li
        print("contact %s added" % name)
    elif not phoneNumberValidator(phone):
        print("Phone number is inValid")
li=addContact(9989794454,"mastan@gmail.com")
addDetails("aaaa",li)
```

```
contact aaaa added
Contact aaaa already exists
```

In [33]:
```
contacts
```

Out[33]:
```
{'name1': [9989794454, 'mastan@gmail.com'],
 'name3': [9989794454, 'mastan@gmail.com'],
 'aaaa': [9989794454, 'mastan@gmail.com']}
```

In [35]:
```
def searchContacts(name):
    if name in contacts:
        print(name)
        print("Phone :",contacts[name][0])
        print("Email :",contacts[name][1])
    else:
        print("%s does not exist " % name)
    return
searchContacts("name1")
```

```
name1
Phone : 9989794454
Email : mastan@gmail.com
```

In [ ]:
```
def updateContact(name,phone):
    if name in contacts:
        contacts[name] = phone
    else:
        print("%s does not exist " % name)
updateContact("mastan","9502304797")
```

In [36]:
```
def importContacts(newContacts):
    contacts.update(newContacts)
    print(len(newContacts.keys())," added successfully")
    return
newContacts = {"name4":[9969445464,"name4@gmail.com"]}
importContacts(newContacts)
```

```
1  added successfully
```

In [37]:
```
contacts
```

Out[37]:
```
{'name1': [9989794454, 'mastan@gmail.com'],
 'name3': [9989794454, 'mastan@gmail.com'],
 'aaaa': [9989794454, 'mastan@gmail.com'],
 'name4': [9969445464, 'name4@gmail.com']}
```

In [39]:
```
# Function to list all contacts
def listAllContacts():
    for contact,info in contacts.items():
        print(contact,"\n","phone: ",info[0],"\n","Email: ",info[1])
listAllContacts()
```

```
name1
 phone:  9989794454
 Email:  mastan@gmail.com
name3
 phone:  9989794454
 Email:  mastan@gmail.com
aaaa
 phone:  9989794454
 Email:  mastan@gmail.com
name4
 phone:  9969445464
 Email:  name4@gmail.com
```

In [38]:
```
contacts.items()
```

Out[38]:
```
dict_items([('name1', [9989794454, 'mastan@gmail.com']), ('name3', [9989794454,
'mastan@gmail.com']), ('aaaa', [9989794454, 'mastan@gmail.com']), ('name4', [99
69445464, 'name4@gmail.com'])])
```

In [ ]:
```
# Function to edit Contact info
def editContact(name,phone,email):
    if name in contacts:
```

In [ ]:
```
```

File Handling in python

File - Document containing information residing on the permnent storage
Types - Text, PDF, CSV etc
File IO - Channelling I/O Data to files
Default I/O Channels - keyboard / Screen

Change I/O Channel to files for reading and writing
ReadFile - Input from file
Write to a file - Output a file
Read / Write file - open (filename,mode)

In [51]:
```
# Function to read a file
def readFile(filename):
    f = open(filename,'r')
    fileData = f.read()
    f.close()
    return fileData
filename = "DataFiles/data.txt"
# for line in readFile(filename).split('\n'):
#     print(line)
def printFileDataLines(filename):
    with open(filename,'r') as f:
        for line in f:
            print(line,end="")
    return
printFileDataLines(filename)
# or
print(readFile(filename))
```

```
Line1
Line2
Line3Line1
Line2
Line3
```

In [56]:
```
# Function to write data into a file
def writeIntoFile(filename,filedata):
    with open(filename,'w') as f:
        f.write(filedata)
    return
filename = "DataFiles/data.txt"
writeIntoFile(filename,"new Data")
```

In [62]:
```
# Function to append data to a file
def appendDataToFile(filename,filedata):
    with open(filename,'a') as f:
        f.writelines(filedata)
    return
filename = "DataFiles/data.txt"
filedata = ["line4","line5"]
appendDataToFile(filename,filedata)
```

In [ ]:
```

```