Day Objectives:

- Regular Expression

```
- Constructing Regular Expressions for various use cases
- Regular Expressions Module and related in python
- Improving the Contacts application with name and phone number vari
ous
```

- File Handling
  - Text files
  - Upgrading the contacts Application to store contact

Regular Expressions

- Pattern Matching
- Symbolic Notation of a pattern
  - pattern: Format which repeats
  - pattern(RE) represents The set of all values
- [0-9] ->Any digit
- [a-z] ->Any lower case alphabet
- [2468] ->All single digit multiples of 2
- ^[0-9]{1}$ ->Only single digit numbers
- ^[0-9]{3}$ ->Only three digit numbers
- [0-9]*0$ -> All multiples of 10
- ([1-9][0-9]*[05])|^([5])$ -> All multiples of 5
- ^[1-9][0-9]{9}$ -> All 10 digits numbers

- [w][o][r][d] or (word) ->Specific word is highlights

- ^[6-9][0-9]{9}|$^{[}0][6-9][0-9]9[0-9]9$ -> Validating Phone number(India)(start with 9876 followed by 9 digits)

-[0-9a-z][0-9a-z_.]{4,13}[0-9a-z][@][a-z0-9]{3,18}[.][2-4a-z]{2,4}$ -> EmailValidation(username@domain.extension (mailto:username@domain.extension))

- Username

```
- Length of username : [6,15]
- No special characters other than_.
- should not begin and end with_ .
- character Set : all digits and alphabet
```

- domain

```
- Length of domain  : [3,18]
- No special characters
- Character Set : all digits and alphabet
```

- extension

    - Length of extension : [2,4]
    - No special characters
    - Character Set : alphabet

- ->Any string of length 5 that starts with 'a' and ends with 'z' ^[a]...[z]$[a]. *[z] -> Any sting any lenth start with 'a' and ends with 'z'

In [ ]:

In [39]:
```
#Function  to validate a phone number
import re
def phoneNumberValidator(number):
    pattern = '^[6-9][0-9]{9}$|^[0][6-9][0-9]{9}[0-9]{9}$'
    if re.match(pattern,str(number)):
        return True
    return False
phoneNumberValidator(9652668543)
```

Out[39]: True

In [35]:
```
contacts = {"name1":9876543210}

def addcontacts(name,phone):
    #verify that the contact does not already exist
    if name not in contacts and phoneNumberValidator(phone):
        contacts[name] = phone
        print("contact %s added")
    if name in contacts:
        print("contact %s already exists" %name)
    elif not PhoneNumberValidator(phone):
        print("phone number is valid")
    return

addcontacts("name1","9774462523")
```

```
contact name1 already exists
```

In [22]:
```
import re
def emailValidator(email):
    pattern ='[0-9a-z][0-9a-z_.]{4,13}[0-9a-z][@][a-z0-9]{3,18}[.][2-4a-z]{2,4}$
    if re.match(pattern,email):
        return True
    else:
        return False
emailValidator("nityaas@gmail.com")
```

Out[22]: True

In [18]:
```
contacts = {"name1":[9876543210, "name1@domain.ext"], "name2":[987654345,"name2@
def addcontacts(name,phone,email):
    #verify that the contact does not already exist
    if name  in contacts :
        print(name,"already exists.")
    else:
        if not phoneNumberValidator(phone):
            print("Invalid phone number")
        if not emailValidator(email):
            print("Invalid email address")
            return
        newcontact = []
        newcontact.append(phone)
        newcontact.append(email)
        contacts[name] = newcontact
    return
addcontacts("name3","9774462523", " name3hf@gmailcom")
```

```
Invalid email address
```

In [31]:
```
def searchcontacts(name):
    if name in contacts:
        print(name)
        print("phone :", contacts[name][0])
        print("Email :", contacts[name][0])
    else:
        print("%s does not exists " %name)
    return
searchcontacts("name2")
```

```
name2 does not exists
```

In [23]:
```
def importcontact(newcontacts):
    contacts.update(newcontacts)
    print(len(newcontacts.keys()), "contact added")
    return
newcontacts = {"name2" :[975323454,"name_412@gmail.com"],"name3" :[8634567898, "
importcontact(newcontacts)
contacts
```

```
2 contact added
```

Out[23]:
```
{'name1': [9876543210, 'name1@domain.ext'],
 'name2': [975323454, 'name_412@gmail.com'],
 'name3': [8634567898, 'name13434@gmail.com']}
```

In [60]:
```
#Function to list all contacts
def ListAllcontacts():
    for contact,info in contacts.items():
        print(contact,"\n","phone:", info[0], "\n", "email:" , info[1])
    return

ListAllcontacts()
```

```
name1
 phone: 9876543210
 email: name1@domain.ext
name2
 phone: 975323454
 email: name_412@gmail.com
name3
 phone: 8634567898
 email: name13434@gmail.com
```

In [41]:
```
def editcontact(name,phone,email):
    if name in contacts:
        if phoneNumberValidator(phone) and emailValidator(email):
            contacts["name1"]=[9848158283,'tej123@gmail.com']
        else:
            print('not modified')
    print(contacts)
editcontact("name1",9848158283,"tej123@gmail.com")
```

```
{'name1': [9848158283, 'tej123@gmail.com']}
```

In [ ]:

File Handing in python

File - Document containing information residing on the permanent storage Types - Text, PDF,CSV etc

File I/O - Channelling I/O data to files

Defalut I/O channnels - Keyboard / screen

Change I/O channel to files for Reading and writing

Read a file - Input from file

Write to a file - output to a file

Read/write file - open(filename, mode)

In [45]:
```
#Function to read a file

def readFile(filename):
    f= open(filename,'r')
    filedata = f.read()
    f.close()
    return filedata

filename = "Data files/data.txt"
filedata = readFile(filename)
#for line in filedata.split("\n"):
   # print(line)
#readFile(filename)
def printFileDataLines(filename):
    f = open(filename,'r')
    for line in f:
        print(line, end=" ")
    return

printFileDataLines(filename)
print(readFile(filename))
```

```
Line1
 Line2
 Line3 Line1
Line2
Line3
```

In [50]:
```
# Function to write data into a file

def writeIntoFile(filename,filedata):
    with open(filename,'w') as f:
        f.write(filedata)
    return
filename = 'Data files/data.txt'

writeIntoFile(filename,"new data\n")
```

In [51]:
```
#Function to append data to a file

def appendDataToFile(filename,filedata):
    with open(filename, 'a') as f:
        #for line in filedata:
        f.write(filedata)
    return

filedata = "Line2\nline3"
appendDataToFile(filename,filedata)
```

In [ ]:
```

```