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 using regular expression
- File Handling
  - Text Files
  - Upgrading the Contacts applications to store Info in to a File

In [ ]:

Basics of Requlal Expression

- [0-9]: to check all are numbers or not
- [A-Z]: to check all are upper case or not
- [a-z]: to check all are lower case or not
- ^ : to check starting char
- $: to check ending char
- * : to check from starting to ending
- . : any data

Tasks

Multiples of 5

- ^[0-9]*[05]$

Validating Indian Mobile numbers starting with[6,7,8,9] of 10 digits

- ^([6-9][0-9]{9})|(0)[9876][0 − 9]9)|([+][9][1][6-9][0-9])$

Validating Email ID:

- ^[a-zA-Z][a-zA-Z0-9_.]*(@apssdc.in)$
- ^[0-9a-z][a-z0-9_.]{4,17}[a-z0-9][@][a-z0-9]{3,18}[.][a-z]{2,4}$|
- ^[0-9a-z][a-z0-9_.]{4,17}[a-z0-9][@][a-z0-9]{3,18}[.][a-z]{2,4}[.][a-z]{2,4}$

Dot: if dot executed then it can be anything

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

In [ ]:

Designing of Contact Application

Constraints:

- User Name
  - 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
- Extensions -Length of extension :[2,4]
  - No special charaters
  - Character set = lower case alphabet

In [ ]:

```
```

In [12]:

```
# Function to validate a phone number

import re

def PhoneNumberValidator(num):
    pattern='^([6-9][0-9]{9}$)|^([0][9876][0-9]{9})$|([+][9][1][6-9][0-9])$'
    if re.match(pattern,str(num)):
        return True
    return False
def EmailValidator(Email):
    pattern="^[0-9a-z][a-z0-9_.]{4,17}[a-z0-9][@][a-z0-9]{3,18}[.][a-z]{2,4}$"
    if re.match(pattern,str(Email)):
        return True
    return False


PhoneNumberValidator(8886785229)
EmailValidator("anilkumar_t@apssdc.in")
```

Out[12]:

```
True
```

In [ ]:

```
```

In [15]:

```
contacts={"Anil Peter":[8886785229,"anilkumar_t@apssdc.in"],"John Silva Raju":[7702023300,"johnsilvaraju7@gmail.com"]}
```

In [16]:

```
def addContact(name, phone, email):
    #verify that the contact doesnot already exist
    if name in contacts:
        print(name, "already exists.")
        return
    else:
        if not PhoneNumberValidator(phone):
            print("Invalid Phone number")
            return
        if not EmailValidator(email):
            print("Invalid Email address")
            return
        newcontact = []
        newcontact.append(phone)
        newcontact.append(email)
        contacts[name] = newcontact
        print(name, "added successfully")
    return

addContact("Daddy", 9441448680, "daddy143@anil.com")
```

```
Daddy added successfully
```

In [ ]:

```

```

In [19]:

```
# Searching a contact in contacts Dict
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("Anil Peter")
```

```
Anil Peter
  Phone : 8886785229
  Email : anilkumar_t@apssdc.in
```

In [ ]:

```

```

In [20]:

```
# 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()
```

```
Anil Peter
 Phone : 8886785229
 Email : anilkumar_t@apssdc.in
John Silva Raju
 Phone : 7702023300
 Email : johnsilvaraju7@gmail.com
Daddy
 Phone : 9441448680
 Email : daddy143@anil.com
```

In [ ]:

```

```

In [24]:

```
# Function to edit contact information

def editcontact(name, phone, email):
    if name in contacts:
        contacts["Anil Peter"]: [8074278199, "teegala.anilkumar1@gmail.com"]
        print("%s contact modified sucessfully" % name)
    else:
        print("%s does not exist" % name)
editcontact('Anil Peter', 8074278199, 'teegala.anilkumar1@gmail.com')
```

```
Anil Peter contact modified sucessfully
```

In [ ]:

```

```

File Handling in Python

- **File :** It is a Document containg informationm residing on the permanent storage (Cloud/Local)
- **Types of Files :** Text,Image,PDF,CSV Etc.
- **File I/O :** Channeling I/O data to Files

Default I/O channel for Python Environment is - Keyboard/Screen

Change the I/O channel to files for Reading & Writing into Files

- Read a file - Input to a file
- Write to a file - Output to a file

File Handling Syntax

Read/ Write file - open('File_Name','Mode')

In [26]:

```
# Function to read a file

def ReadFile(filename):
    f=open(filename,'r')
    filedata=f.read()

    f.close()
    #for x in f.readline():

    return filedata

for line in ReadFile(input("Enter a File name to read data")).split('\n'):
    print(line)



def PrintDatainLines(FileName):
    f=open(FileName,'r')
    for line in f:
        print(line,end='')
    f.close()
    return
PrintDatainLines("DataFiles/First.txt")
```

```
Enter a File name to read dataDataFiles/First.txt
Name     :     Anil Peter
Role     :     Trainer cum Developer
Organisation     :     APSSDC

Name     :     Anil Peter
Role     :     Trainer cum Developer
Organisation     :     APSSDC
```

In [ ]:

```

```

In [72]:

```
# Finction to Append data into a File

def AppendDataintoFile(FileName,FileData):
    l=[]
    with open(FileName,'w') as f:
        for keys,values in FileData.items():
            p=keys,':',values
            for line in p:
                f.writelines(line)
                f.write("     ")
            f.write('\n')
    return

filename='DataFiles/First.txt'


filedata={'Name':'Anil Peter','Role':'Trainer cum Developer','Organisation':'APSSDC'}
AppendDataintoFile(filename,filedata)
```

In [ ]:

```

```