Day Objectives:

- Regular Expressions
  - Constructing Regular Expressions for various use cases
  - Regular Expressions Module and Related in python
  - improving contact application
- File Handling
  - Text Files
- - Upgrading the contacts Application to store contact information in a text file

Regular Expression

- pattern matching
- Symbolic notation of a pattern

```
    - pattern: Format which repeats
    - pattern(RE) represents the set of all strings matches that patte
rn
```

- [0-9]->any digit
- [a-z]->any lower case alphabet
- -> All single digits multiples of 2

[0-9].....>in between take any value

- `^[0-9]` .....>statr only in betenn no
- `^[0-9]{1}` .....>take only one no in between that
- `[0-9]+` .....>takes no split by space
- `[0-9]{3}` .....>only 3 digits
- `[0-9]*8$` .....>all multiples of 10
- `^([1-9][0-9]*[05])|([5])$` .....>multiples of 5
- `^([0-9]){9}` .....>all 10 digit numbers
- `[a][b][c][d]` .....>search word abcd or `(abcd)`
- `[6-9][0-9]{9}$|^[0][6-9][0-9]{9}|^[+][9][1][6-9][0-9]{9}$` .....>phone no validation
- `^[0-9a-z][0-9a-z_.]{4,13}[0-9a-z][@][a-z0-9]{3,18}[.][a-z]{2,4}$` .....>gmail,validation

```
- username
        - length of uname:[6,15]
        - no special characters others than _
        - should not begin and end with _
        - characterset :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 character
        - character set:alphabet
```

- Any string of length 5 and start with a and end withz
  - ^[a]...[z]. . . . . >. . . $means it takes any values 3 in b/w - [a]. *[z].....>it is also takes any values

In [4]:
```
import re
def phonevalidate(num):
    pattern='^[6-9][0-9]{9}$|^[0][6-9][0-9]{9}|[+][9][1][6-9][0-9]{9}$'
    if re.match(pattern,str(num)):
        print("valid no")
    else:
        print("notnvalid")
    return
phonevalidate(9676047715)
```

```
valid no
```

File Handling in python

- file...>document containing information residing
- types..>text,pdf,csv.etc
- file i\o ....>channeling i/o data to files
- default i/o channels:::keyboard /screen
- change i/o channel to files for reading and writing
- read file -->input from file
- write to a file...>output to a file
- Read/write file--->open(filename,mode)

In [5]:
```
#function to read file
def readfile(filename,mode):
    f=open(filename,mode)

    print(f.read())
readfile("DataFiles/file1.txt","r")
```

```
Welcome satheesh
goodEvening satheesh
are you single
yes ohhhh!!!!
why dude
?
```

In [6]:
```
#print line wise
def printFileDatalines(filename):
    f=open(filename,'r')
    for line in f:
        print(line)
    return
printFileDatalines("DataFiles/file1.txt")
```

```
Welcome satheesh

goodEvening satheesh

are you single

yes ohhhh!!!!

why dude

?
```

In [7]:
```
def printFileDatalines(filename):
    with open(filename,'r') as f:
        for line in f:
            print(line,end=" ")
        return
printFileDatalines("DataFiles/file1.txt")
```

```
Welcome satheesh
 goodEvening satheesh
 are you single
 yes ohhhh!!!!
 why dude
 ?
```

In [9]:
```
#write data into file
def writedata(filename,filedata):
    with open(filename,'w') as f:
        f.write(filedata)
    return
filedata=input()
filename="./DataFiles/file2.txt"
writedata(filename,filedata)
```

```
satheesh
```

In [13]:
```
#append data to file
def appendDataFile(filename,filedata):
    with open(filename,'a') as f:
#         for line in filedata:
            f.write('\n'+filedata)
    return
filedata=input()
filename="./DataFiles/file2.txt"
appendDataFile(filename,filedata)
```

```
welcome vind
```

In [ ]:
```

```