Search Function

Ethans
Learn from experts

```
import re

line = "this is the string contain sentence"
matchObj = re.search("This is the string Match", line, re.I)

if matchObj:
   print "\n matchObj.group() : \n", matchObj.group()
   print "\n matchObj.group(1) : \n", matchObj.group(1)
else:
   print "No match!!"
```


Findall Function

Ethans
Learn from experts

- It scan the complete string and get all occurrences of a given pattern within a string
- The re.findall function returns a list of all matches.

- Help on match function:

```
>>> import re
>>> help(re.findall)
```

Parameter	Description
pattern	The regex/pattern which need to match
string	This is the string, where re need to be matches
flags	Modifiers or Flags to support more functionality to re

Findall Function ex:

Ethans
Learn from experts

```
import re

line = "this is the string contain sentence"
allMatches = re.findall("This", line, re.I)

if len(allMatches):
   print "\n allMatches : \n", allMatches
else:
   print "No match!!"
```


The First Wildcard

Ethans
Learn from experts

Wildcards (are also called as quantifiers) are the operator symbols which have specific meaning inside regular expression.

For example: . (Dot or period) matches any character, digit, alphanumeric character except newline character (\n).

```
def refind(pat, str):
   match = re.search(par, str):
   if match:
      print match.group()
   else:
      print 'Pattern not found'
```

```
>>> refind('a.b', 'this is acb string')
acb
>>> refind('a.b', 'this is jatbin string')
atb
>>> refind('a.b', 'this is a b string')
a b
>>> refind('a.b', 'this is a\tb string')
a	b
```

Match Operator Itself

Ethans
Learn from experts

In many cases, user may wants to match the operator symbol itself in the regular expression. We can suppress the wild cards and special characters itself by backslash (\)

Ex:

```
>>> refind('10.', 'this is a number 101 and ip address 10.')
101
>>> refind('10\.', 'this is a number 101 and ip address 10.')
10.
```


Other Wildcards

Ethans
Learn from experts

These wildcard characters do not matches themselves. Until and unless they suppressed by backslash.

Following are the other wildcards:

Wildcard	Meaning
*	matches Zero or more occurrence of previous character/s
+	matches One or more occurrence of previous character/s
?	matches Zero or One occurrence of previous character/s

Other Wildcards

Ethans
Learn from experts

REGEX	Matches
AbC*	It matches A followed by b followed by either Zero or more occurrence of C. i.e. Ab, AbC, AbCCCC. AbCCCCCCCCCCC
AbC+	It matches A followed by b followed by minimum one or more occurrence of C i.e. Abc, AbCCCCCCCC, AbCCC
AbC?	It matches A followed by b followed by one or Zero occurenece of C. i.e. Ab, AbC
Ab(cd)*	It matches A followed by b followed by either Zero or more occurrence of cd i.e. Ab, Abcd, Abcdcd
Ab(cd)+	It matches A followed by b followed by minimum one or more occurrence of cd i.e. Abcd, Abcdcd
Ab(cd)?	It matches A followed by b followed by either one or zero occurrence of cd i.e. Abcd, Ab

Slide 211

www.ethans.co.in

Combine Wildcards

Ethans
Learn from experts

REGEX	Matches
Ab+C*	It matches A followed by minimum one or more occcurenece of b followed by either Zero or more occurrence of C. i.e. Ab, Abc, Abbccc. Abbccccccccccc
A.C+	It matches A followed by any character followed by minimum one or more occurrence of C i.e. AZc, Azccc. AEccccccccccc
..C?	It matches any two characters followed by b followed by one or Zero occurenece of C. i.e. Ab, AbC
<.*>	It matches anything inside tags <> i.e. <HTML>, <TAGS>
\(.+\)	It matches minimum one character inside brackets cd i.e. (Abcd), (a)
ab+c?	It match a followed by one or more b followed by zero or one c. i.e. "abbbbc" or "abc", but not "ac"

Slide 212

www.ethans.co.in

Character Class

Ethans
Learn from experts

REGEX	Matches
[abc]	It matches any string which has either 'a' or 'b' or 'c'
[abcdefghijklmnopqrstuvwxyz]	It matches any string which has either 'a' or 'b' or 'c' or so on till 'z'
[a-z]	It matches any string which has either 'a' or 'b' or 'c' or so on till 'z'
[0-9]	It matches any string which has 0 or 1 or 2 or 3 till 9
[a-zA-Z0-9]	It matches any string which has characters from a-z and A-Z and 0-9
[a-z_]	It matches any string which has characters from a-z or _ (underscore)

Character Class

Ethans
Learn from experts

REGEX	Matches
[^abc]	It matches any string which has neither 'a' nor 'b' nor 'c'
[^abcdefghijklmnopqrstuvwxyz]	It matches any string which has neither 'a ' or 'b' or 'c' or so on till 'z'
[^a-z]	It matches the string which has neither 'a' or 'b' or 'c' or so on till 'z'
[^aeiou]	It matches the string which has no vowels.
[IL][^abc]	It matches the string has 'I' or 'L' should not followed by 'a' nor 'b' nor 'c'
[^a-z_]	It matches the string doesn't have a-z or _ (underscore)

Character Class

Ethans
Learn from experts

REGEX	Matches
[aA][0-9]+	It matches any string which has 'a' or 'A' followed by any number and occurrence can any number of times.
A+.[.?]	It matches any string which has 'A' any number of times followed by any character followed by either '.' or '?'
a[bc]	It matches any string which has 'a' followed by either 'b' or 'c'
A[abc]?	It matches the string which has 'A' followed by zero or one occurrence of either 'a' or 'b' or 'c'
[a-z_.]\@	It matches the string has 'a' to 'z' or '_' or '.' followed by '@'

Shortcuts

Ethans
Learn from experts

Shortcut	Say	Meaning
\s	Any space, tab or new line characters	[\t\n]
\S	Other than space, tab or newline character	[^\t\n]
\d	Any digit	[0-9]
\D	Other than digit	[^0-9]
\w	Digits, characters or _ (underscore)	[a-zA-Z0-9_]
\W	Other than digit, character or _	[^a-zA-Z0-9_]

Objective – Module 9

Ethans
Learn from experts

Database Interface

- Creating a Database with SQLite 3
- CRUD Operations,
- Creating a Database Object.
- Python MySQL Database Access
- DML and DDL Operations with Databases
- Performing Transactions
- Handling Database Errors
- Disconnecting Database

DB Interface with Python

Ethans
Learn from experts

Python Database modules supports a wide range of database servers:

- IBM DB2
- Firebird
- Informix
- Ingres
- MySQL
- Oracle
- PostgreSQL
- SAP DB
- Microsoft SQL Server
- Microsoft Access
- Sybase
- Teradata
- IBM Netezza

DB Interface with Python -2

Ethans
Learn from experts

Standard for RDBMS Databases

Python DB API/drivers/modules provides conventional approach while working with databases using Python standard structures and syntax.

This includes the following:

- Importing the Python-DB module.
- Acquire a connection with the database.
- Create a DB cursor
- Issuing SQL statements and stored procedures.
- Closing the connection

SQLite Database

Ethans
Learn from experts

- SQLite is a relational database management system contained in a C programming library.
- SQLite is a popular choice as embedded database software for local/client storage in application software such as web browsers.
- It is fast,reliable and the entire DB is stored in single disk file.
- Python module sqlite3 intended to work with SQL Lite3.

Source: https://en.wikipedia.org/wiki/SQLite

Acquire DB connection

Ethans
Learn From experts

- Python sqlite3 module will be used to connect SQLite3 Database
- It is default available along with the Python installable.
- connect, close are the functions in below example

```
>>> # Importing sqlite3 DB
>>> import sqlite3
>>> # Creating a DB in RAM
>>> dbr = sqlite3.connect(':memory:')
>>> # Creating/Using a DB in/from a file
>>> db = sqlite3.connect('ethans')
>>> # Closing a DB connection
>>> db.close()
```


Create table

Ethans
Learn From experts

```
import sqlite3 # Importing module
db = sqlite3.connect('ethans')
dbCursor = db.cursor() # DB cursor
dbCursor.execute(""" CREATE TABLE
    employee(id int, name text,dept text,
    salary int) """)
db.close()
```

```
>>> help(dbCursor.execute)
execute(...)
  Executes a SQL statement.

>>> dir(dbCursor)
['arraysize', 'close', 'connection', 'description', 'execute', 'executemany',
'executescript', 'fetchall', 'fetchmany', 'fetchone', 'lastrowid', 'next',
'row_factory', 'rowcount', 'setinputsizes', 'setoutputsize']
```

Insert Data - 1

Ethans
Learn From experts

```
''' Insert a record '''

import sqlite3 # Importing module

db = sqlite3.connect('ethanstech')
dbCursor = db.cursor() # DB cursor

dbCursor.execute(""" INSERT INTO employee
    values (1, 'ethans','training', '50000')
    """)

db.commit()
db.close()
```


Insert Data - 2

Ethans
Learn From experts

```
''' Insert a record '''

import sqlite3 # Importing module

db = sqlite3.connect('ethanstech')
dbCursor = db.cursor() # DB cursor

dbCursor.execute(""" INSERT INTO employee
    values (?, ?, ?, ?)""",
    (1, 'ethans','training', 50000))

db.commit()
db.close()
```

Insert Data - 3

Ethans Learn from experts

```
''' Insert a record '''

import sqlite3 # Importing module

db = sqlite3.connect('ethanstech')
dbCursor = db.cursor() # DB cursor

eid, name, dept, salary = 1, 'ethans', 'training', 50000

dbCursor.execute(""" INSERT INTO employee
    values (?, ?, ?, ?)""",
    (eid, name, dept, salary))

db.commit()
db.close()
```


Insert Data - 4

Ethans Learn from experts

```
import sqlite3, sys
db = sqlite3.connect('Employee')
cursor = db.cursor()

id, name, dept, salary = 6, 'Chetan', 'IT', 10000
sql ='''INSERT INTO emp values (%d, '%s', '%s', %d)''' \
     %(id, name, dept, salary)

cursor.execute(sql)
db.commit()

cursor.execute('''Select * from emp''')
for record in cursor.fetchall():
    print record

db.close()
```

Insert Data - 5

Ethans
Learn From experts

```
''' Insert a record '''

import sqlite3 # Importing module

db = sqlite3.connect('ethanstech')
dbCursor = db.cursor() # DB cursor

eid, name, dept, salary = 1, 'ethans', 'training', 50000

dbCursor.execute(""" INSERT INTO employee
    values (:id, :name, :dept, :salary)""",
    {'name':name, 'id':eid, 'dept':dept, 'salary':salary})

db.commit()
db.close()
```


Insert Data - 6

Ethans
Learn From experts

```
''' Insert a record '''

import sqlite3 # Importing module

db = sqlite3.connect('ethanstech')
dbCursor = db.cursor() # DB cursor

eid, name, dept, salary = 1, 'ethans', 'training', 50000

dbCursor.executemany(""" INSERT INTO employee
    values (?,?,?,?)""",
    [(eid, name, dept, salary), (2, 'Steve', 'training', 60000)])

db.commit()
db.close()
```

Insert and Select Data

Ethans Learn from experts

```
import sqlite3, sys
db = sqlite3.connect('Employee')
cursor = db.cursor()

id, name, dept, salary = 7, 'Santosh', 'IT', 10000
cursor.execute('''INSERT INTO emp values (:id, :name, :dept, :sal)'''
               {'sal':salary, 'name':name, 'dept':dept, 'id':id})
db.commit()

cursor.execute('''Select * from emp where dept = ?''', ('IT',))
for record in cursor.fetchall():
    print list(record)

db.close()
```


Insert using executemany and select

Ethans Learn from experts

```
import sqlite3, sys
db = sqlite3.connect('Employee')
cursor = db.cursor()

id1, name1, dept1, salary1 = 8, 'Nitin', 'IT', 10000
id2, name2, dept2, salary2 = 9, 'Bob', 'IT', 20000

dataset = [(id1, name1, dept1, salary1), (id2, name2, dept2, salary2)]
cursor.executemany('''INSERT INTO emp values (?, ?, ?, ?)''', dataset)
db.commit()

cursor.execute('''Select * from emp where dept = ?''', ('IT',))
for record in cursor.fetchall():
    print list(record)

db.close()
```

Drop table

Ethans
Learn from experts

```
''' Drop a table '''

import sqlite3 # Importing module

db = sqlite3.connect('ethanstech')
dbCursor = db.cursor() # DB cursor

dbCursor.execute(""" DROP table employee""")

db.close()
```


Objective – Module 10

Ethans
Learn from experts

Introduction to python modules

- Introduction to Python repository
- What are modules and how to locate?
- Install modules
- What is pip, easy_install and source forge?
- Install modules using pip and easy install
- Python compiler for windows
- XLS modules
- Web scraping using Python
- Important modules

Py Repository

Ethans
Learn from experts

- Python Package Index or PyPI is the official third-party software repository for the Python programming language. Python developers intend it to be a comprehensive catalog of all open source Python packages
- While the PyPI website is maintained by the Python Software Foundation, its contents are uploaded by individual package maintainers. Python package managers such as pip default to downloading packages from PyPI.
- Location: https://pypi.python.org/pypi/
- Access: http://pypi-ranking.info/alltime to know the ranking of all packages

Source: Wiki

What are tools to install modules

Ethans
Learn from experts

There are many ways to install and manage Python packages from PyPI.

- **easy_install** - https://pythonhosted.org/setuptools/easy_install.html
- **Pip** - http://docs.python-guide.org/en/latest/starting/install/win/
- **Setup** - https://docs.python.org/2/install/
- **Windows Installation (Unofficial)** http://www.lfd.uci.edu/~gohlke/pythonlibs
- **Source Forge** - https://sourceforge.net/projects/numpy/
- **MySQL Example** – https://pypi.python.org/pypi/MySQL-python/1.2.5#downloads

Easy Install

Ethans
Learn from experts

- easy_install is a python module that lets you download, build, install, and manage Python packages automatically.
- easy_install default available in python >= 2.7.9 version
- To install easy_install, download the ez_setup.py file from https://pypi.python.org/pypi/setuptools
- To install a package using easy_install, only supply the filename or URL of a source distribution or .egg
- On Windows, an easy_install.exe launcher will also be installed, so that you can just type easy_install as long as it's on your PATH. If typing easy_install at the command prompt doesn't work, check to make sure your PATH includes the appropriate C:\\Python2X\\Scripts directory.

easy_install commands

Ethans
Learn from experts

- To get the help page
  easy_install --help
- Install any package by name – Search PyPI for the latest version and download it. Build and install the package manually.
  easy_install xlrd
- Install a package via URL
  easy_install -f https://pypi.python.org/pypi/xlwt
  easy_install http://example.com/path/to/MyPackage-1.2.3.tgz
- To upgrade a package
  easy_install "SomePackage==2.0"
  easy_install "SomePackage>2.0"
  easy_install --upgrade beautifulsoup
- To uninstall the package:
  easy_install -mxN PackageName

pip

Ethans
Learn from experts

- pip is a package management system used to install and manage software packages written in Python. Many packages can be found in the Python Package Index (PyPI).
- Python 2.7.9 and later (on the python2 series), and Python 3.4 and later include pip (pip3 for Python 3) by default.
- pip is a recursive acronym that can stand for either "Pip Installs Packages" or "Pip Installs Python".
- pip is quite similar to easy_install. It is a python module that lets you download, build, install, and manage Python packages automatically.
- To install pip, download the get-pip.py file from https://pip.pypa.io/en/latest/installing.html
- To install the package using pip, supply the filename only.

pip commands

Ethans
Learn from experts

The most common scenario to install a package using pip is:

- pip –h # to display the help page
- pip list # to display the installed packages
- pip list –outdated # to display all outdated package
- pip search xlrd # to search the package in repo
- pip install Package # latest version
- pip install Package==1.0.4 # specific version
- pip install 'Package>=1.0.4' # minimum version requirement
- pip uninstall package # to uninstall the package
- pip install –upgrade # to upgrade the package

Interact with spreadsheets

Ethans
Learn from experts

There are so many python packages available to work with Excel files that will run on any Python platform and that do not require either Windows or Excel to be used. They are fast, reliable and open source:

openpyxl - for reading and writing Excel 2010 files (ie: .xlsx)

xlsxwriter - An alternative package for writing data, formatting information and, in particular, charts in the Excel 2010 format (ie: .xlsx)

xlrd - This package is for reading data and formatting information from older Excel files (ie: .xls)

xlwt - This package is for writing data and formatting information to older Excel files (ie: .xls)

xlutils - This package collects utilities that require both xlrd and xlwt, including the ability to copy and modify or filter existing excel files.

Read data from xlrd

Ethans
Learn from experts

```
import xlrd
myWorkBook = xlrd.open_workbook('sampleFile.xlsx')
for sheet in myWorkBook.sheets():
    print 'Sheet:',sheet.name
    for row in range(sheet.nrows):
        for col in range(sheet.ncols):
            print "cell: row - %d col - %d and value
- %r " %(row, col,
sheet.cell(row,col).value)
```