

PYTHON FUNDAMENTALS

Curs interactiv de python

STRUCTURA SEDINTA 6 : MODULE AVANSAT

▀ Tema sedinta anterioara

▀ Module avansat in Python

- Tema :

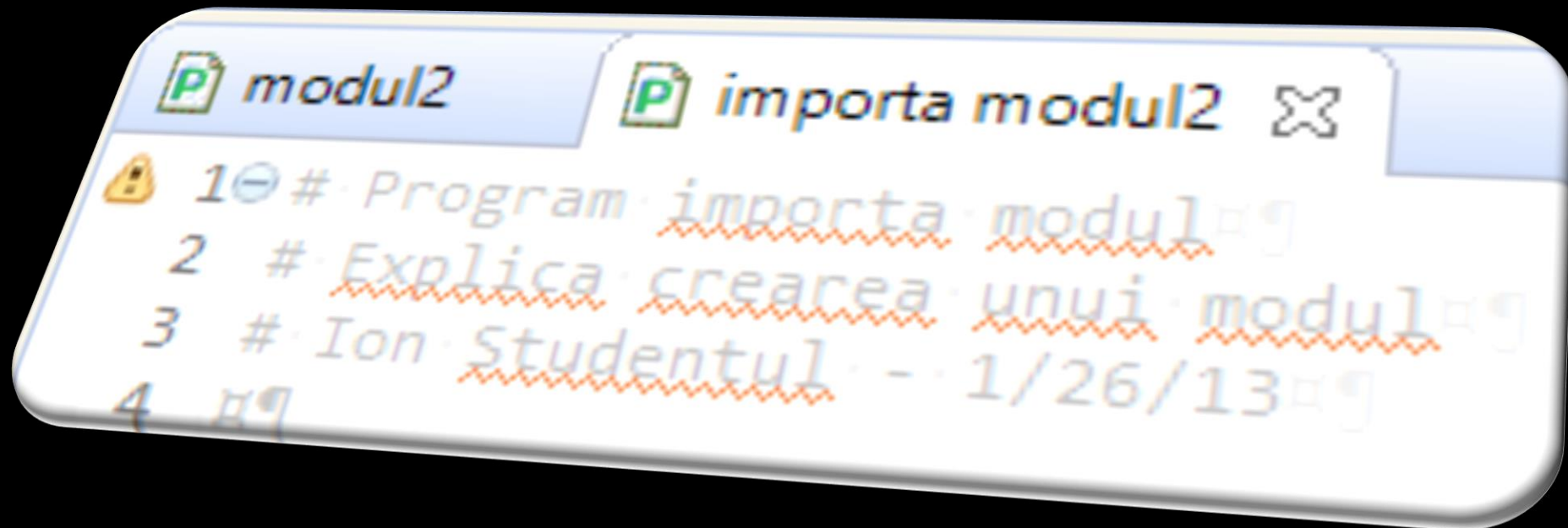
Creati un modul ce sa contina o functie numita Adunare ce va primi doi parametri si ii va aduna daca poate (try-else) returnand rezultatul. In cazul in care nu poate sa adune/concateneze cele doua elemente sa returneze un text explicativ. Adaugati un if `__name__ == "__main__"` cu un text explicativ la rularea directa.

Creati un fisier care sa importe sys si os.

In cazul in care sistemul de operare este windows (sys.platform) sa importe modulul altfel sa returneze un mesaj si iasa din program (if-else)

Acest program trebuie sa afiseze calea curenta apoi sa o schimbe calea curenta la calea "C:\Program Files". Confirmati schimbarea. Rulati apoi functia Adunare pentru "2" si "3" (sir de caractere).

TEMA SEDINTA ANTERIOARA



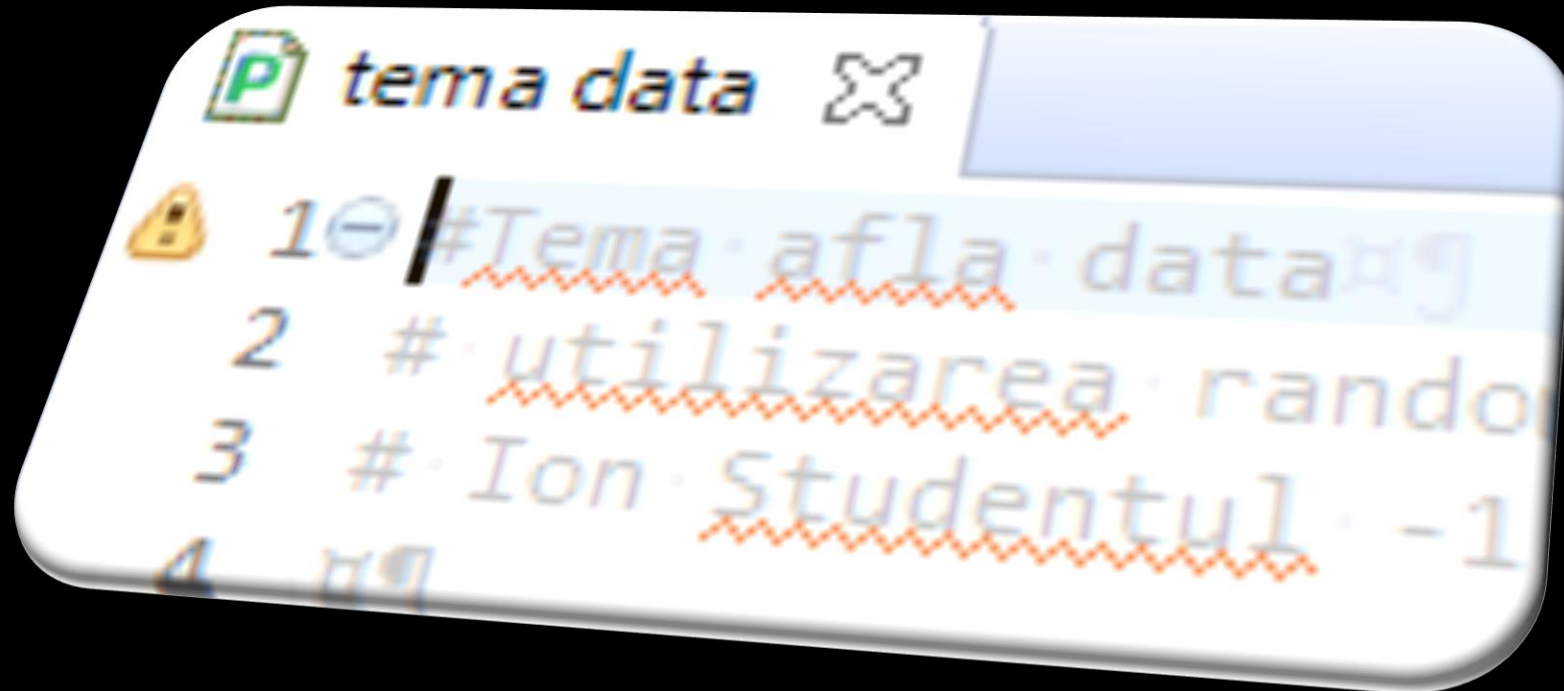
```
1 # Program: importa modul  
2 # Explica crearea unui modul  
3 # Ion Studentul - 1/26/13  
4
```

- Tema:

Creati un program care sa aiba o functie numita dataAleatoriu ce returneaza o lista din trei elemente. Primul element al listei sa fie un an aleatoriu intre 2000 si 2020. Al doilea element al listei sa fie o luna aleatorie returnata ca numar intre 1 si 12. Al treilea element al listei sa fie o zi intre 1 si 28.

Apelati de trei ori functia data si stocati rezultatul intr-o variabila. Aflati ziua din saptamana a datelor returnate.

TEMA SEDINTA ANTERIOARA



```
1 #Tema afla data
2 # utilizarea random
3 # Ion Studentul -1
4 %%
```

EXPRESII REGULATE: MODULUL RE

```
>>> x = "Trei culori cunosc pe lume: rosu bourbon,  
galben ocru si turcuaz"  
>>> x.find("rosu")  
28  
>>> x.find("Error")  
-1  
>>>
```

EXPRESII REGULATE: MODULUL RE

```
>>> x = " Warning:error on reading file! "  
>>> x.find("warning")  
-1  
>>> x.find("Error")  
-1  
>>>
```


EXPRESII REGULATE: MODULUL RE

O expresie regulata este o secventa speciala de caractere care te ajuta să găsești într-un șir de caractere sau un set de șiruri de caractere un tipar dat

.	punct	Se potrivește cu orice caracter.
\d		Se potrivește cu orice caracter zecimal
\D		Se potrivește cu orice caracter non-zecimal
\s		Se potrivește cu caracterul space
\S		Se potrivește cu orice caracter ce nu e space
\w		Se potrivește cu orice caracter alfanumeric.
\W		Se potrivește cu orice caracter non-alfanumeric

EXPRESII REGULATE: MODULUL RE

Mecanisme repetitive:

* - (steluta) repeta de 0 sau mai multe ori un caracter sau un șir de caractere.

Ex: `sc*apa` va avea potrivire și cu `sapa(c=0)`, `scapa(c=1)`, `sccapa(c=2)`...etc

Daca dorim să testăm un șir de caractere care să se repete de cate ori vrem atunci folosim paranteze rotunde. `o(ma)*` va avea potrivire și cu `o (ma=0)`, și cu `oma(ma=1)` și cu `omama(ma=2)` etc.

+ - (plus) Acesta repeta de 1 sau mai multe ori un caracter sau un șir de caractere.

EXPRESII REGULATE: MODULUL RE

Mecanisme repetitive:

? - (semanul intrebarii) are rolul de a testa repetitia unui caracater sau unui şir de caracater de 0 sau o singura data.

{m,n} - unde m este de cate ori trebuie minimum să se repete un caracater sau un şir de caractere, iar n reprezinta numarul maxim de repetari. Dacă am avea expresia regulata $c1\{1,2\}b$ acesta s-ar potrivi doar cu $c1b$ şi $c11b$.

EXPRESII REGULATE: MODULUL RE

Variatiuni:

. * - zero sau mai multe caractere (oricare).

dacă dorim să cautam un caracter special, trebuie să folosim caracterul \ care anuleaza acel caracter special ca în orice șir de caractere unde doream să utilizam " sau '

EXPRESII REGULATE: MODULUL RE

doua functii ale modulului :

- `re.match(pattern, string)`

`pattern`- șir de caractere pentru potrivire

`string`- în ce șir să caute.

Returneaza `None` daca nu gaseste nimic.

`re.match()` cauta doar la inceputul unui sir de caractere nu pe fiecare linie in cazul in care cautam pe mai multe linii.

OBS: A se utiliza in conjunctie cu `try-else`.

EXPRESII REGULATE: MODULUL RE

doua functii ale modulului :

- `re.search(pattern, string)`

`pattern`- șir de caractere pentru potrivire

`string`- în ce șir să caute.

Returneaza `None` daca nu gaseste nimic.

`re.search()` cauta pe fiecare linie in cazul in care cautam pe mai multe linii.

OBS: A se utiliza in conjunctie cu `try-else`.

MODULUL RE: MATCH

```
>>> sir = """acasa la aceste persoane academice."""
>>> potrivire = re.match("(ac\\w+)", sir)
>>> if potrivire:
    potrivire.groups()

('acasa',)
```

MODULUL RE: MATCH

```
>>> sir = """Am fost acasa la aceste persoane academice."""
>>>
>>> potrivire = re.match("(ac\\w+)", sir)
>>> if potrivire:
>>>     potrivire.groups()

>>> print potrivire
None
```

MODULUL RE: SEARCH

```
>>> sir = """Am fost acasa la aceste persoane academice."""  
>>> cauta=re.search("(ac\\w+)", sir)  
>>> print cauta.groups()  
( 'acasa', )
```

MODULUL RE: GROUPS() VERSUS GROUP(0)

Daca nu caut intre paranteze groups() nu o sa returneze ceva coerent

```
>>> x = re.search("ac(\w)*",sir)
>>> x.group(0)
'acasa'
>>> x.group(1)
'a'
>>> x.groups()
('a',)
>>> x = re.search("(ac(\w)*)",sir)
>>> x.groups()
('acasa', 'a')
>>> x.group(0)
'acasa'
>>>
```

MODULUL RE: EXAMPLE

Daca cautam de mai multe ori acelasi tipar avem doua solutii: ciclarea

```
>>> sir = """Am fost acasa la aceste persoane academice."""
>>> lista = sir.split()
>>> print lista
['Am', 'fost', 'acasa', 'la', 'aceste', 'persoane', 'academice.']
>>> for elem in lista:
    cauta=re.search("(ac\\w+)", elem)
    if cauta :
        print cauta.groups()

('acasa',)
('aceste',)
('academice',)
```

MODULUL RE: EXEMPLE

Daca cautam de mai multe ori acelasi tipar avem doua solutii: findall

```
>>> sir = """Am fost acasa la aceste persoane academice."""
>>> x = re.findall("ac(\w)*",sir)
>>> x
['a', 'e', 'e']
>>> sir = """Am fost acasa la aceste persoane academice."""
>>> x = re.findall("(ac(\w)*)",sir)
>>> x
[('acasa', 'a'), ('aceste', 'e'), ('academice', 'e')]
>>> x = re.findall("ac\w*",sir)
>>> x
['acasa', 'aceste', 'academice']
>>>
```


MODULUL RE: EXAMPLE

Programul de mai jos introduce elemente noi cum ar fi:

- `stdin, stdout = os.popen2("COMANDA CMD")`
- *Creez doua obiecte ce vor avea scop scrierea unei comenzi (obiectul `stdin`) si citirea textului returnat (obiectul `stdout`). Manipularea se face cu ajutorul metodelor intalnite la fisiere (`close`, `read`, `readlines` etc.)*
- *Pentru a aplica comanda: `stdin.close()`*
- *Pentru a citi textul returnat: `stdout.read()`*

MODULUL RE: EXEMPLE

```
real_re X
1 import os, re
2 print("aplicam comanda ping in consola")
3
4 stdin, stdout = os.popen2('ping -n 1 8.8.8.8')
5 stdin.close()
6
```

MODULUL RE: SPLIT

Atributul **split** imparte un șir de caractere pe baza unui tipar dat, rezultatul fiind returnat într-o listă

Iată sintaxă pentru split:

- `re.split(pattern, string)`
- `pattern`- șir de caractere pentru potrivire
- `string`- în ce șir să caute.

MODULUL RE: SPLIT

```
# Program RE 2
# Explica regular exp
# Ion Studentul - 1/26/13

import re

sir = """Aseara am mancat un hotdog mare."""

m = re.split("(d\\w+)", sir)
if m:
    print m
```

Acest modul imparte un sir de caractere pe baza unui tipar dat, rezultatul fiind returnat intr-o lista.

Console   Debug

<terminated> D:\Catalin\Predare Python\carte\cap 5\split_re.py
['Aseara am mancat un hot', 'dog', ' mare.']

MODULUL RE: SUB

Atributul sub inlocuieste un sir de caractere pe baza unui tipar dat
`re.sub(pattern,replace, string)`

- pattern- şir de caractere pentru potrivire
- replace – ce inlocuim
- string- cu ce inlocuim.

MODULUL RE: SUB

```
>>> a = """ata  
asa  
ara"""  
>>> inloc = re.sub("a.a", "info", a)  
>>> print inloc  
info  
info  
info  
>>>
```


MODULUL NTPLIB

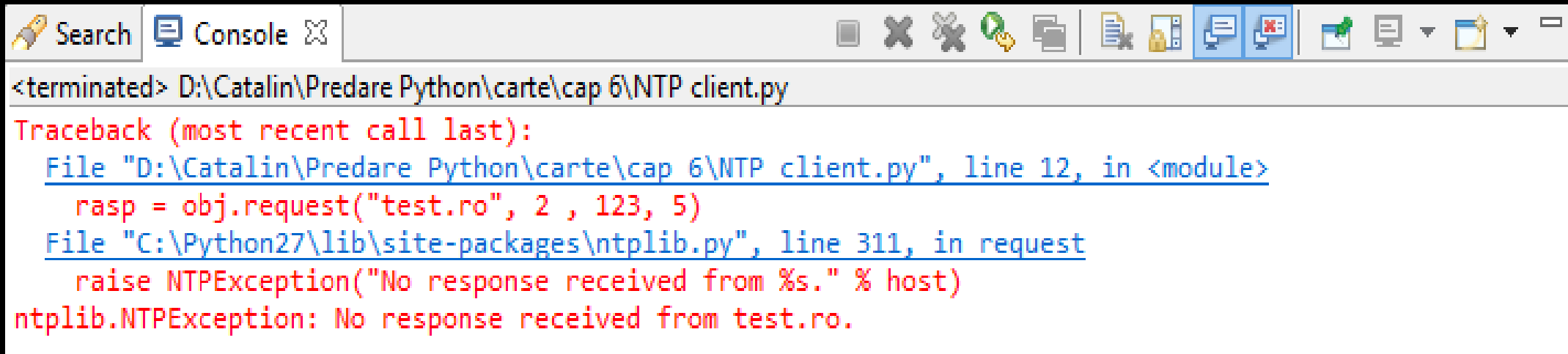
- NTP este un serviciu de retea ce are ca scop sincronizarea ceasului diferitelor dispozitive de retea
- <https://pypi.python.org/packages/source/n/ntplib/ntplib-0.3.2.tar.gz>

```
D:\Catalin\Predare Python\carte\Sedinta 6\Programe\NTPLIB\ntplib-0.3.2\ntplib-0.3.2>setup.py install
running install
running build
running build_py
running install_lib
copying build\lib\ntplib.py -> C:\Python27\Lib\site-packages
byte-compiling C:\Python27\Lib\site-packages\ntplib.py to ntplib.pyc
running install_egg_info
writing C:\Python27\Lib\site-packages\ntplib-0.3.2-py2.7.egg-info
```

MODULUL NTPLIB

- Cererea catre serverul NTP contine patru parametrii:
- host – Aici este ntp3.usv.ro. Reprezinta serverul ntp de la care solicitam informatia.
- Versiunea – Aici este utilizată varianta 2.
- Port – Aici este utilizat port no. NTP standard, adica 123.
- Timeout – Aici este utilizată valoarea 5. Reprezinta dupa cat timp va genera un mesaj de timeout.

Daca serverul nu raspunde returneaza Eroare.



The screenshot shows a Python console window with a toolbar at the top containing icons for search, console, and various file operations. The main area displays a traceback error. The path to the script is 'D:\Catalin\Predare Python\carte\cap 6\NTP client.py'. The error originates from line 12 of the script, where a request is made to 'test.ro' with a timeout of 5 seconds. The traceback points to line 311 of the 'ntplib.py' file in the 'C:\Python27\lib\site-packages' directory, where an 'NTPException' is raised because no response was received from 'test.ro'.

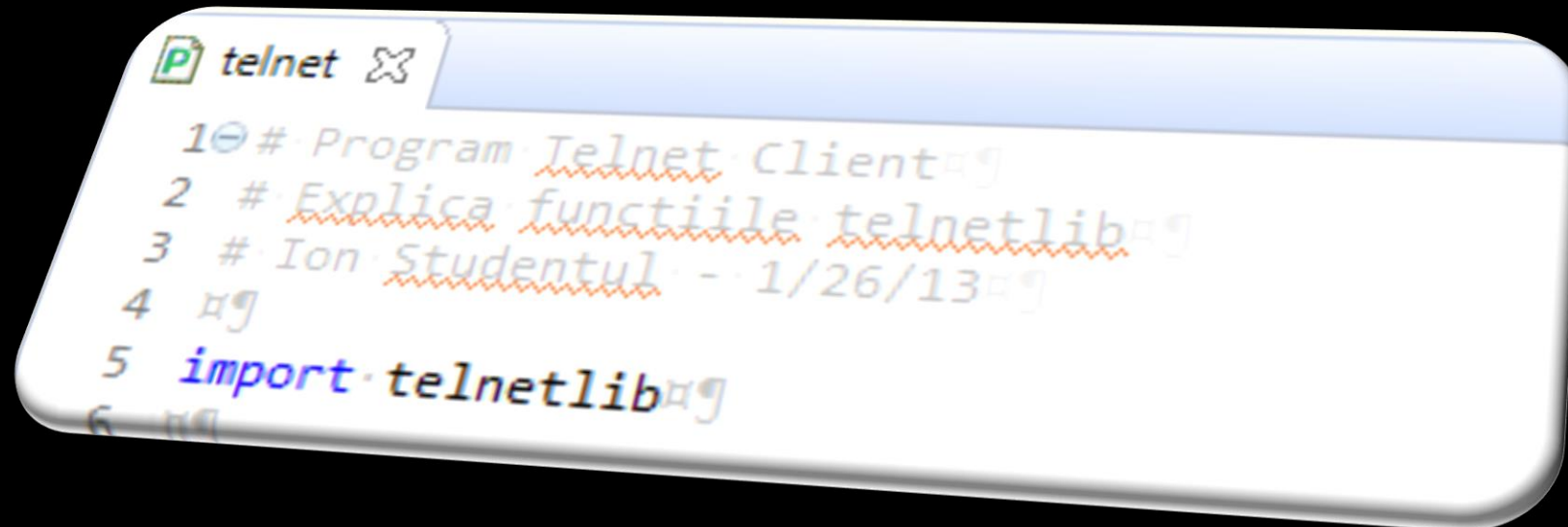
```
<terminated> D:\Catalin\Predare Python\carte\cap 6\NTP client.py
Traceback (most recent call last):
  File "D:\Catalin\Predare Python\carte\cap 6\NTP client.py", line 12, in <module>
    rasp = obj.request("test.ro", 2 , 123, 5)
  File "C:\Python27\lib\site-packages\ntplib.py", line 311, in request
    raise NTPException("No response received from %s." % host)
ntplib.NTPException: No response received from test.ro.
```

MODULUL NTP

INFOACADEMY.NET

```
NTP client ✖  
1 # Program NTP Client  
2 # Explica functiile ntplib  
3 # Ion Studentul -- 1/26/13  
4  
5 #importam modulele utilizate  
6 import ntplib  
7 import time
```

- TELNET este un serviciu de retea ce are ca scop conectarea la distanta cu diferite dispozitive de retea
- Este un modul standard



```
1 # Program Telnet Client
2 # Explica functiile telnetlib
3 # Ion Studentul - 1/26/13
4
5 import telnetlib
6
```

MODULUL FTP

- FTP este un serviciu de retea ce are ca scop transferul de fișiere la distanță cu diferite dispozitive de rețea
- Este un modul standard
- Pentru a crea un obiect `ftplib` să apelăm comanda: `ftp = ftplib.FTP('HOST')`. Așa cum ați ghicit `HOST` poate fi o adresă IP sau numele dispozitivului. Prin apelarea [ftp.login\(\)](#) se realizează conectarea la serverul `ftp` declarat la crearea obiectului.
- Cu ajutorul comenzii `ftp.retrlines('list')` extragem fișierele din serverul `ftp`.
- Dacă dorim să închidem conexiunea cu serverul `ftp` și să ștergem obiectul `ftp` atunci apelăm `ftp.close()`.

MODULUL FTP

```
>>> import ftplib
>>> ftp = ftplib.FTP('192.168.1.50')
>>> ftp.login()
'230 Anonymous user logged in.'
>>> ftp.retrlines('LIST')
01-13-06  02:42AM                28521 Blue hills.jpg
08-19-14  06:48PM                <DIR>         poze
01-13-06  02:16AM                71189 Sunset.jpg
01-13-06  02:54AM                83794 Water lilies.jpg
01-13-06  02:24AM               105542 Winter.jpg
'226 Transfer complete.'
>>> ftp.close()
>>>
```

Extrageți fișierele server-ului cu Adresa IP scrisă pe tablă

MODULUL FTP

```
ftp_py1  ftp_py2
1 # Program ftp Client
2 # Explica functiile telnetlib
3 # Ion Studentul - 1/26/13
4
5 import ftplib
6
7 ftp = ftplib.FTP('192.168.1.5')
8
```

Tip de cerere	Explicatie
GET	Metoda GET înseamnă prelua orice informație este identificata prin cerere. Dacă cererea se referă la un procedeu de producere a datelor, va fi returnat ca entitatea unui raspuns și nu textul sursă a procesului.
POST	<p>Metoda POST este folosita pentru a solicita ca serverul să accepte informatii incluse în cerere. POST este conceput pentru a permite o metodă uniformă pentru a acoperi următoarele funcții:</p> <ul style="list-style-type: none">- Adnotarea resurselor existente;- Postarea unui mesaj la un avizier, newsgroup, lista de discutii, sau un grup similar de articole;- Extinderea unei baze de date printr-o operație de adăugare. <p>Metoda POST depinde de cerere deoarece foloseste URL-ul pentru a transmite informatia dorita de la utilizator catre server.</p>

Tip de cerere	Explicatie
HEAD	Metoda HEAD este identic cu GET cu excepția faptului că serverul nu trebuie să returneze un corp mesaj (message-body) în raspuns. Informatia continuta în antetele HTTP (header) trebuie să fie identica cu informațiile trimise ca răspuns la o solicitare GET. Aceasta metoda poate fi utilizată pentru obținerea de informatie despre pagina web implicata fără a transfera body-ul paginii în sine. Această metodă este deseori folosită pentru testarea link-urilor pentru a determina valabilitatea, accesibilitatea, sau modificarea recenta.
OPTIONS	Metoda OPȚIUNI reprezintă o cerere pentru informații despre opțiunile de comunicare disponibile pentru cereri sau raspunsuri. Această metodă permite clientului pentru a determina optiunile și / sau cerintele asociate cu o resursă, sau cu caracteristicile unui server, fără a presupune o acțiune de solicitare a resurselor.
DELETE	Metoda DELETE sterge o resursa alocata pe serverul web
PUT	Metoda PUT este similara cu POST în sensul ca transmite informatie catre server în vederea prelucrării și stocării. Totusi aceasta metoada difera fata de POST în sensul ca PUT transfera efectiv fisierul selectat în cadrul unui URL catre server pentru a putea fi accesat pe server chiar și de aplicatii externe.

HTTP

Putem să determinăm dacă pagina a fost extrasă cu succes sau ce tip de eroare am întâmpinat după codul răspuns (status-code). Iată cele mai uzuale coduri întâlnite:

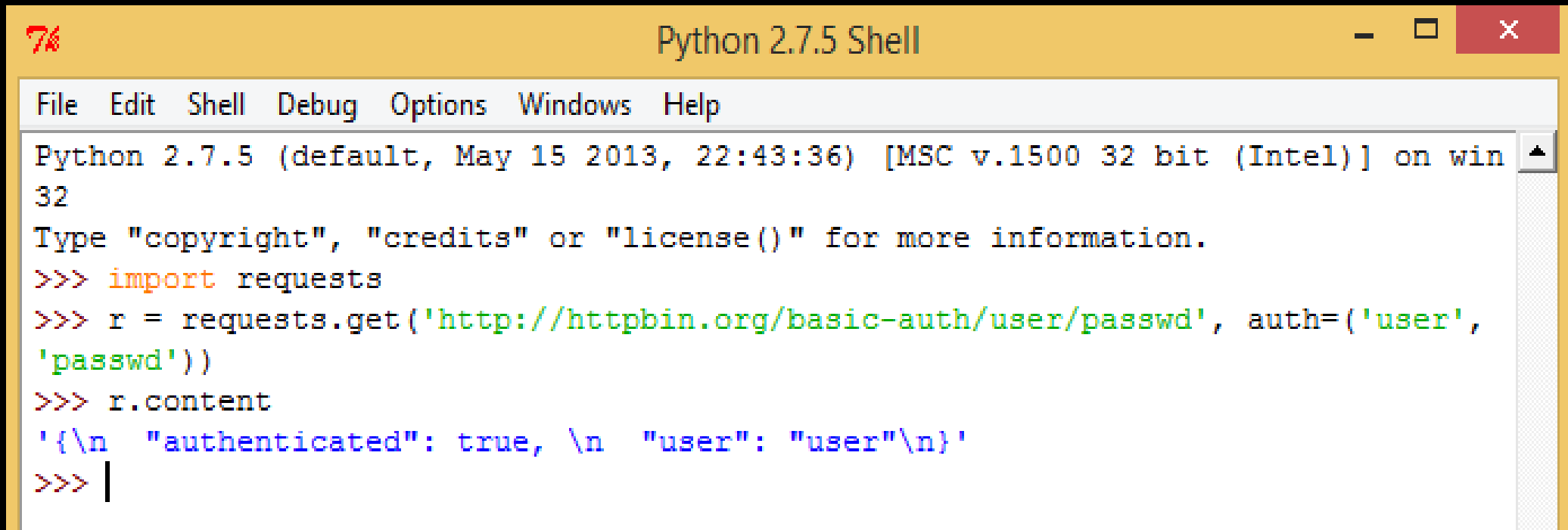
- 200 – Pagina a fost încărcată cu succes.
- 403 – Accesul este interzis (FORBIDDEN)
- 404 – Fișierul nu a fost găsit
- 408 – Cererea a expirat (request timeout)
- 307 – Pagina a fost temporar redirecționată (temporary_redirect)
- 500 – INTERNAL_SERVER_ERROR
- 301 – MOVED_PERMANENTLY

Python 2.7.5 Shell

```
'<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="ro"><head><meta content="/image
title><script>(function(){window.google={kEI:\'VhT1U7HfGMjXyQP6uoLoAQ\',kEXPI:\'4791,17259,4000116,4003510,40
258,4011679,4012149,4012373,4013414,4013591,4013605,4013723,4013823,4013967,4013979,4014016,4014431,4014636,4
6127,4016309,4016373,4016824,4016976,4017204,4017285,4017595,4017639,4017658,4017659,4017694,4017818,4017894,
40500,4010000,4010000,4010000,4010000,4010014,4010010,4010110,4010101,4010101,4010007,4010100,4010100,4010100,4010100
```

HTTP

Chiar dacă în exemplu de mai sus am utilizat URL-ul celebrului motor de cautare google, vom utiliza o alta pagina ce este special creata pentru testarea programelor de tip web browser, și anume <http://httpbin.org/> .



```
Python 2.7.5 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> import requests
>>> r = requests.get('http://httpbin.org/basic-auth/user/passwd', auth=('user',
'passwd'))
>>> r.content
'{"\n  "authenticated": true, \n  "user": "user"\n}'
>>> |
```

```
>>> r.status_code
200
>>> if (r.status_code == requests.codes.ok):
    print "pagina a fost incarcata cu succes!"

pagina a fost incarcata cu succes!
>>> requests.codes.INTERNAL_SERVER_ERROR
500
>>> requests.codes.REQUEST_TIMEOUT
408
```



```
>>> r = requests.post("http://httpbin.org/post")
>>> r.status_code
200
>>> r = requests.put("http://httpbin.org/put")
>>> r.status_code
200
>>> r = requests.delete("http://httpbin.org/delete")
>>> r.status_code
200
>>> r = requests.head("http://httpbin.org/get")
>>> r.status_code
200
>>> r = requests.options("http://httpbin.org/get")
>>> r.status_code
200
```

HTTP

- Poti extrage și antetele raspuns ale serverelor (headers) ca un dictionar, deci foarte usor de integrat. Dacă antetul cautat nu este gasit în acel dictionary (deci nu este returnat de catre server) atunci va returna None.

```
>>> r.text
u'{\n  "authenticated": true, \n  "user": "user"\n}'
>>> r.headers
{'content-length': '46', 'server': 'gunicorn/18.0', 'connection': 'keep-alive',
'access-control-allow-credentials': 'true', 'date': 'Wed, 20 Aug 2014 22:48:45 G
MT', 'access-control-allow-origin': '*', 'content-type': 'application/json'}
>>> r.headers.keys()
['content-length', 'server', 'connection', 'access-control-allow-credentials', '
date', 'access-control-allow-origin', 'content-type']
```

- Endcoding-ul este un set de reguli ce leaga anumite caractere vizibile pentru om în date transmise pe retea. Dacă folosești orice în afara de caracterele de baza necesare pentru Engleza, utilizatorii s-ar putea să nu poată citi text-ul dacă nu specifice ce codificare ai folosit.

De exemplu: intenția ta e ca text-ul să arate :

Author: Guðrún Guðmundsdóttir. Title: Introduction to character encoding (文字符号化入門). Copyright © 2004-2007 W3C® (MIT, ERCIM, Keio).

dar s-ar putea să fie afisate ca:

Author: GuÃ°rÃ°n GuÃ°mundsdÃ°ttir. Title: Introduction to character encoding (æ—†â—ç¬|â—œ—â…¥é—€). Copyright Â© 2004-2007 W3CÂ® (MIT, ERCIM, Keio).

```
>>> print r.encoding
utf-8
>>> r.encoding = 'ISO-8859-1'
>>> print r.encoding
ISO-8859-1
>>> r.text
u'{"message":"Hello there, wayfaring stranger. If you\'re reading this then you
probably didn\'t see our blog post a couple of years back announcing that this A
PI would go away: http://git.io/17AR0g Fear not, you should be able to get what
you need from the shiny new Events API instead.", "documentation_url":"https://de
veloper.github.com/v3/activity/events/#list-public-events"}'
```

HTTP



Command Prompt - c:\Python27\python.exe -m SimpleHTTPServer 50005



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\PopescuFam>c:\Python27\python.exe -m SimpleHTTPServer 50005
Serving HTTP on 0.0.0.0 port 50005 ...
127.0.0.1 - - [21/Aug/2014 15:05:09] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [21/Aug/2014 15:05:10] "GET /http_files/image001.jpg HTTP/1.1" 200 -
127.0.0.1 - - [21/Aug/2014 15:05:10] code 404, message File not found
127.0.0.1 - - [21/Aug/2014 15:05:10] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [21/Aug/2014 15:05:10] code 404, message File not found
127.0.0.1 - - [21/Aug/2014 15:05:10] "GET /favicon.ico HTTP/1.1" 404 -
192.168.1.50 - - [21/Aug/2014 15:08:24] "GET / HTTP/1.1" 200 -
192.168.1.50 - - [21/Aug/2014 15:08:25] "GET /http_files/image001.jpg HTTP/1.1" 200 -
192.168.1.50 - - [21/Aug/2014 15:08:25] code 404, message File not found
192.168.1.50 - - [21/Aug/2014 15:08:25] "GET /favicon.ico HTTP/1.1" 404 -
```

TRIMITEREA UNUI EMAIL:SMTPLIB

Simple Mail Transfer Protocol (SMTP) este un protocol care se ocupa de transmiterea de e-mailuri și forwardarea de emailuri inter serverele de email.

Python ofera modulul smtplib, ce defineste un client de SMTP sub formă unei sesiuni obiect ce poate fi utilizat pentru a trimite e-mail-uri catre orice masina de pe internet cu SMTP.

Iata o sintaxă simplă pt. a crea un obiect de SMTP care poate fi utilizat ulterior pentru trimiterea unui e-mail:

```
import smtplib
```

```
smtpObj = smtplib.SMTP( [host [, port [, local_hostname]]] )
```

TRIMITEREA UNUI EMAIL:SMTPLIB

Un obiect SMTP are o metoda numita sendmail, care este folosita pentru a trimite mesaje e-mail. Aceasta primeste trei parametrii:

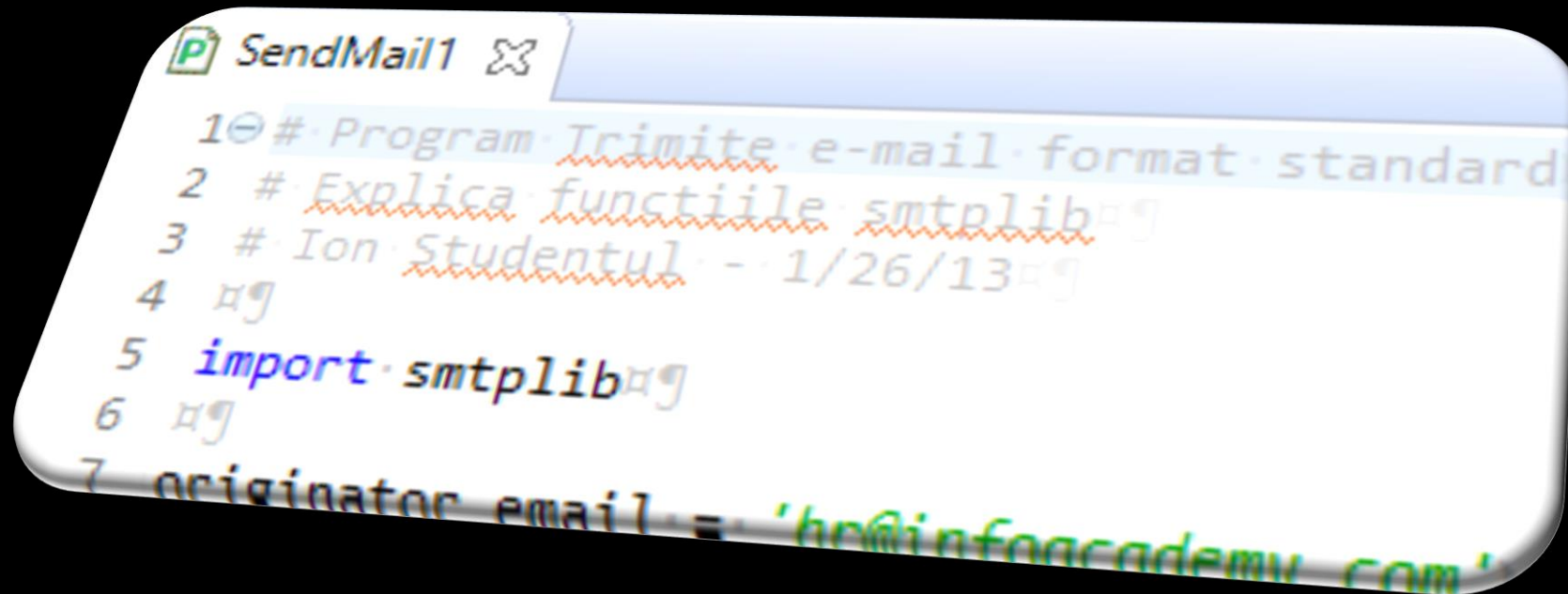
Adresa de trimitere – Un șir de caractere care ofera adresa de trimitere.

Adresa destinatie – Un șir de caractere care ofera adresa de destinatie.

Mesajul- Un mesaj sub formă de șir de caractere ce trebuie să aiba un format special conforme cu anumite RFC-uri.

TRIMITEREA UNUI EMAIL:SMTPLIB

INFOACADEMY.NET



```
1 # Program Trimite e-mail format standard
2 # Explica functiile smtplib
3 # Ion Studentul - 1/26/13
4
5 import smtplib
6
7 originator_email = 'hr@infoacademy.com'
```

TRIMITEREA UNUI EMAIL:SMTPLIB

INFOACADEMY.NET

Instructiuni Google legate de Standarde

(SMTP) Server - necesita TLS or SSL: smtp.gmail.com

Foloseste autentificare:DA

Port pt. TLS/STARTTLS: 587

Port pt. SSL:465

Timp de expirare al serverului : mai mare de 1 minut, recomandat 5 minute

Account Name sau User Name:adresa de email intreaga: test@gmail.com

Email Address: adresa de email intreaga: test@gmail.com

Password: parola Gmail

TRIMITEREA UNUI EMAIL:SMTPLIB

La exemplul anterior vom adauga urmatoarele:

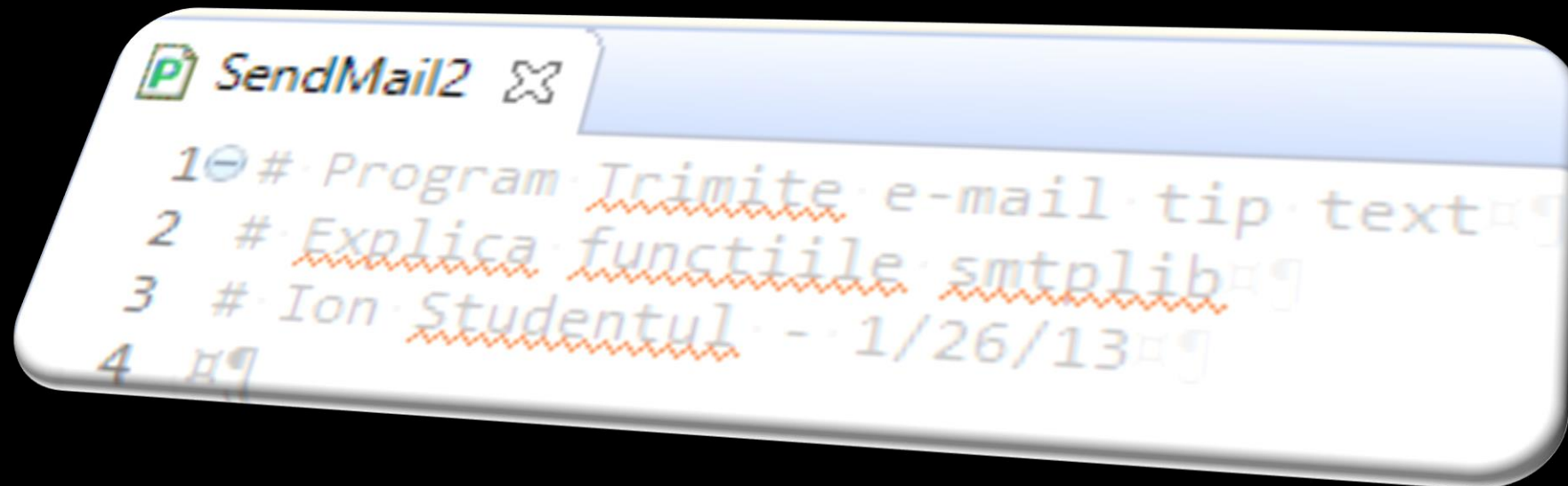
- Indicam și un port la acel server.
- Google utilizeaza criptare prin TLS(predecesorul acestuia este SSL utilizat default pentru https)- este un algoritm de criptare pentru a se asigura ca mesajul trimis nu este citit sau modificat pe parcurs, Mai multe detalii gasiti la link-ul:

http://ro.wikipedia.org/wiki/Transport_Layer_Security

- Apelam metoda login a obiectului creat pentru a ne autentifica la serverul google

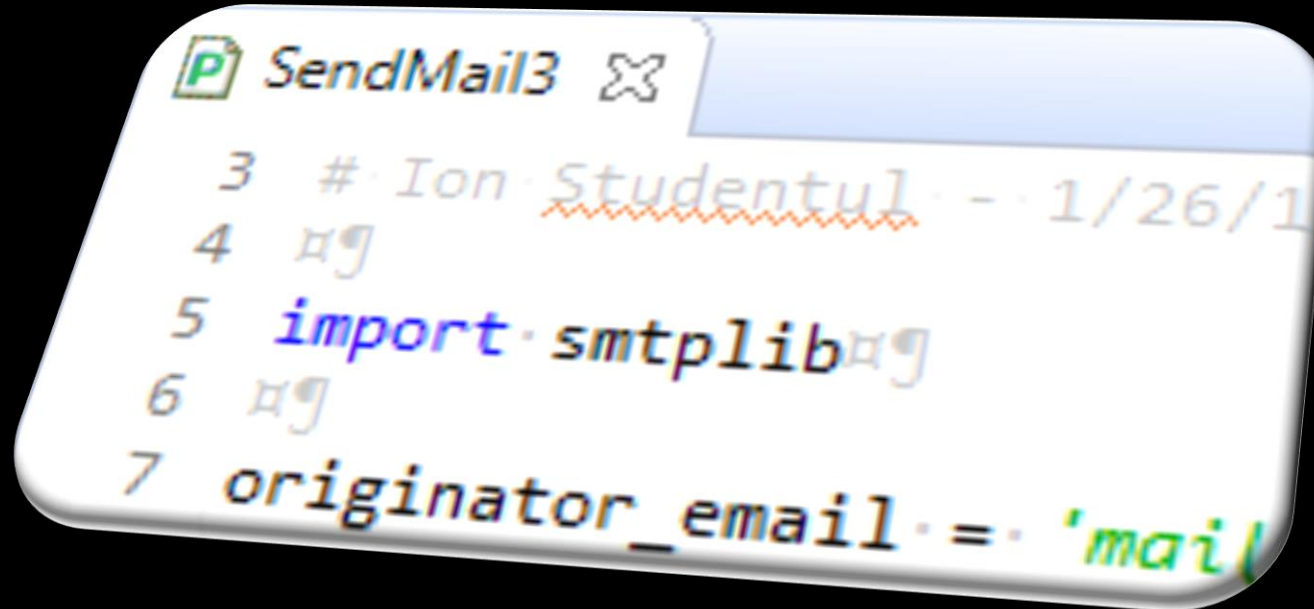
TRIMITEREA UNUI EMAIL:SMTPLIB

INFOACADEMY.NET



```
1 # Program Trimite e-mail tip text  
2 # Explica functiile smtpplib  
3 # Ion Studentul - 1/26/13  
4
```

E-MAIL CARE SĂ FIE DE FORMĂ UNUI HTML



```
3 # Ion Studentul - 1/26/1
4
5 import smtplib
6
7 originator_email = 'mail'
```

DESENAREA DE GRAFICE

- Pentru desenarea de grafice putem utiliza mai multe solutii, dar una din cele mai bune și flexibile solutii este pygal. PyGal este un modul aditional ce trebuie instalat, deci mai jos se regaseste un link de unde puteti descarca modulul:
- <https://pypi.python.org/packages/source/p/pygal/pygal-1.5.0.tar.gz>
- Necesita setuptools in prealabil:

Windows (simplified)

For Windows without PowerShell 3 or for installation without a command-line, download [ez_setup.py](#) using your preferred web browser or other technique and "run" that file.

DESENAREA DE GRAFICE


- PyGal utilizeaza crearea de grafice direct sub formă de fisier imagine de tip SVG.
- Scalable Vector Graphics (SVG) ("grafica vectoriala proportionabila") este un limbaj pentru descrierea de imagini 2D folosind XML. Este un standard al organizatiei W3C a cărui proiectare a început în anul 1999. Permite definirea imaginilor prin 3 metode: text, grafică vectoriala și "bitmap-uri" (fișiere în formatul BMP).
- Deși există aplicații specializate pentru crearea și editarea de SVG-uri, în acest scop poate fi folosit orice editor text. Vizualizarea unei imagini SVG poate fi realizată cu orice browser, deci și integrarea graficelor într-un site este ușor de realizat.

DESENAREA DE GRAFICE

Plot1_pygal

```
1 # Program generare grafic pygal
2 # Explica functiile pygal
3 # Ion Studentul - 1/26/13
4
5 import pygal
6
7 import pygal
```

DESENAREA DE GRAFICE: MAI MULTE RANDURI

 Plot2_pygal ✖

```
1 # Program generare grafi  
2 # Explica functiile pyga  
3 # Ion Studentul - 1/26/1  
4 #
```

DESENAREA DE GRAFICE: TITLU LABEL SI HUMAN_READABLE

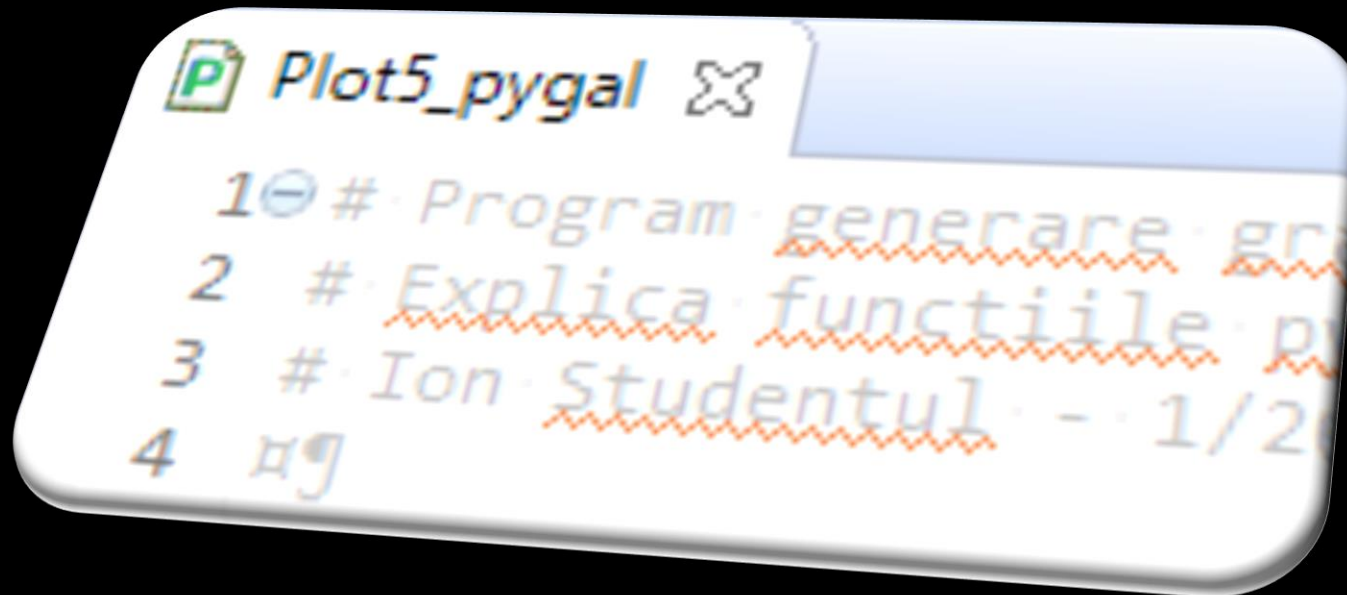
Plot3_pygal

```
1 # Program generare grafic pygal
2 # Explica functiile pygal
3 # Ion Studentul - 1/26/13
4
5 import pygal
```

DESENAREA DE GRAFICE: LINE

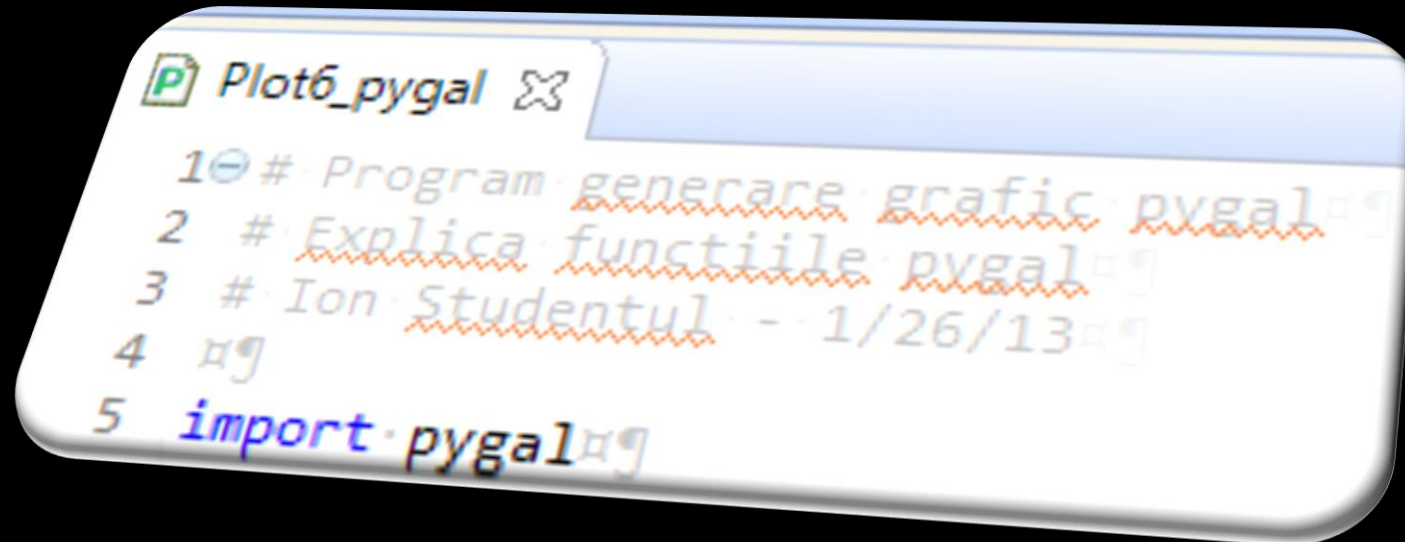
```
Plot4_pygal ✖  
1 # Program generare grafic pygal  
2 # Explica functiile pygal  
3 # Ion Studentul - 1/26/13  
4  
5 import pygal  
6  
7 bar chart = pygal.Line()
```

DESENAREA DE GRAFICE: STACKEDBAR



```
1 # Program generare grafic  
2 # Explica functiile plot  
3 # Ion Studentul - 1/20  
4 #
```


DESENAREA DE GRAFICE: WORLDMAP



```
Plot6_pygal ⌵  
1 # Program generare grafic pygal  
2 # Explica functiile pygal  
3 # Ion Studentul - 1/26/13  
4  
5 import pygal
```

DESENAREA DE GRAFICE: WORLDMAP

INFOACADEMY.NET

<http://pygal.org/documentation/>



TEMA 1

- Modificati programul Tema_clasa1.py primit pe e-mail pentru a rula in consola o comanda de afisare a datei curente. Extrageți cu ajutorul RE anul current. Folosti doar o metoda



Exemple Python

```
C:\Users\Fam.Popescu>date /t  
Fri 01/16/2015
```

TEMA 2

- Realizati un program care sa extraga timpul de la serverul ntp <<2.ro.pool.ntp.org>>. Transformati timpul intr-un format prietenos cu ajutorul time.ctime(). Transmiteti prin e-mail acest timp catre adresa personala gmail/yahoo utilizand o adresa de e-mail gmail/yahoo. Utilizati transmiterea de mesaj ca sir de caractere.
- Dupa testare si reusita inlocuiti parola utilizata la adresa de email cu sirul de caractere <<****>>.

TEMA 3

- Creati un program care sa se conecteze la telnet la adresa <<route-views.routeviews.org>> cu un debug level setat la zero(standard este zero) printr-o variabila definite la inceputul programului. Daca aceasta variabila e diferita de 0 atunci afisati si fiecare linie ce contine metoda
- read_until() aplicata obiectului telnetlib. Astfel creati un debug level
- Conexiunea are nevoie de credentiale prin urmare ar trebui sa utilizeze
- userul: rviews
- parola: ""
- Aplicati apoi comanda <<show version>>
- Aceasta comanda returneaza tabelul de rutare. Extrageți doar pana la intalnirea expresiei <<---more--->>
- Cautati in textul returnat de comanda << show version >>
- sirul de caractere << Version xx.x(x) >> unde x este un numar cu ajutorul re.search() si afisati cu ajutorul metodei group(0).

VA MULTUMESC PENTRU PARTICIPARE

La revedere!