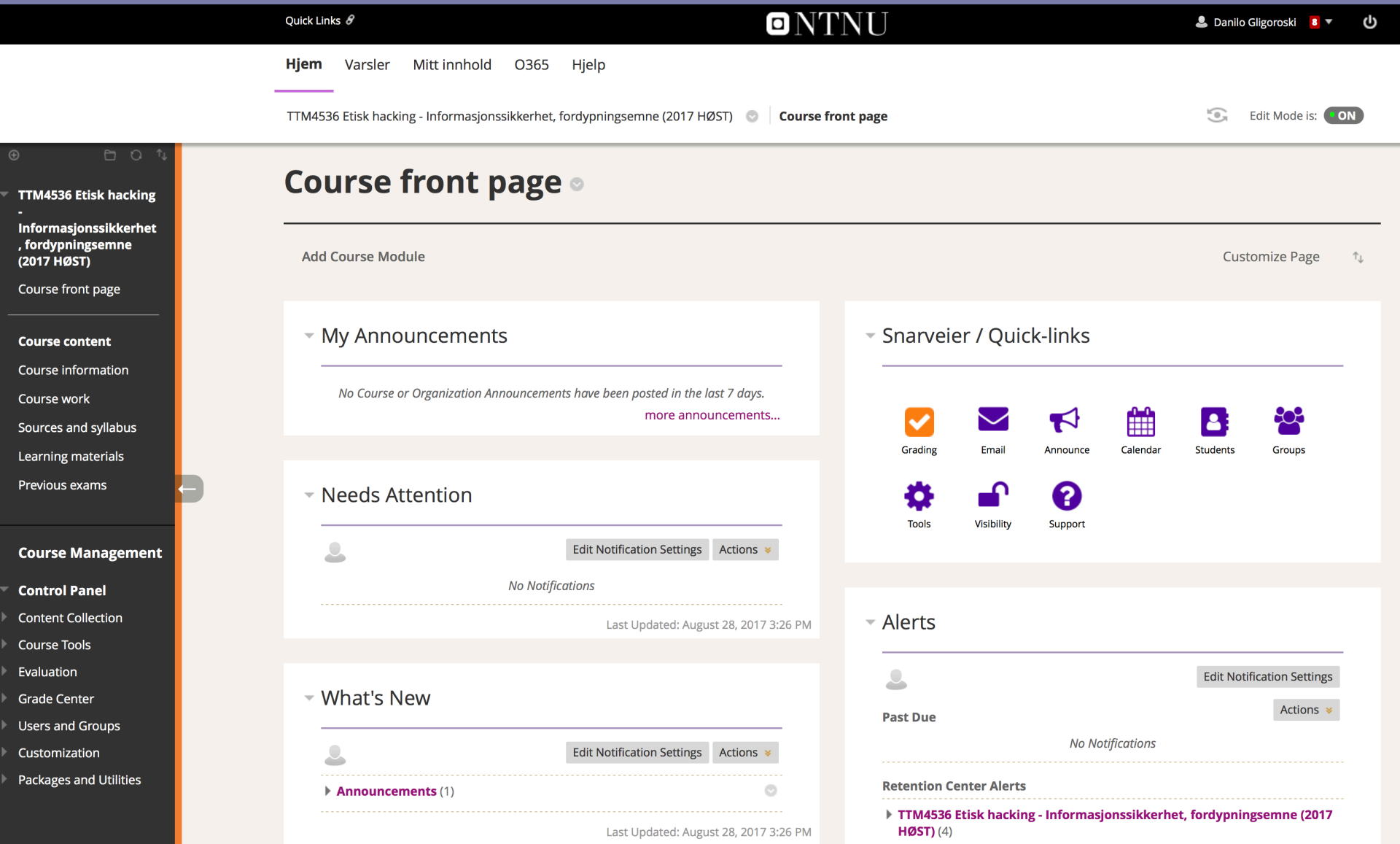
TTM4536 - Ethical Hacking - Information Security, Specialization Course

- Responsible professor: Danilo Gligoroski

- Please visit Blackboard web page:
- Visit and register

Blackboard course web page



Textbook:

Black Hat Python

Python Programming for Hackers and Pentesters

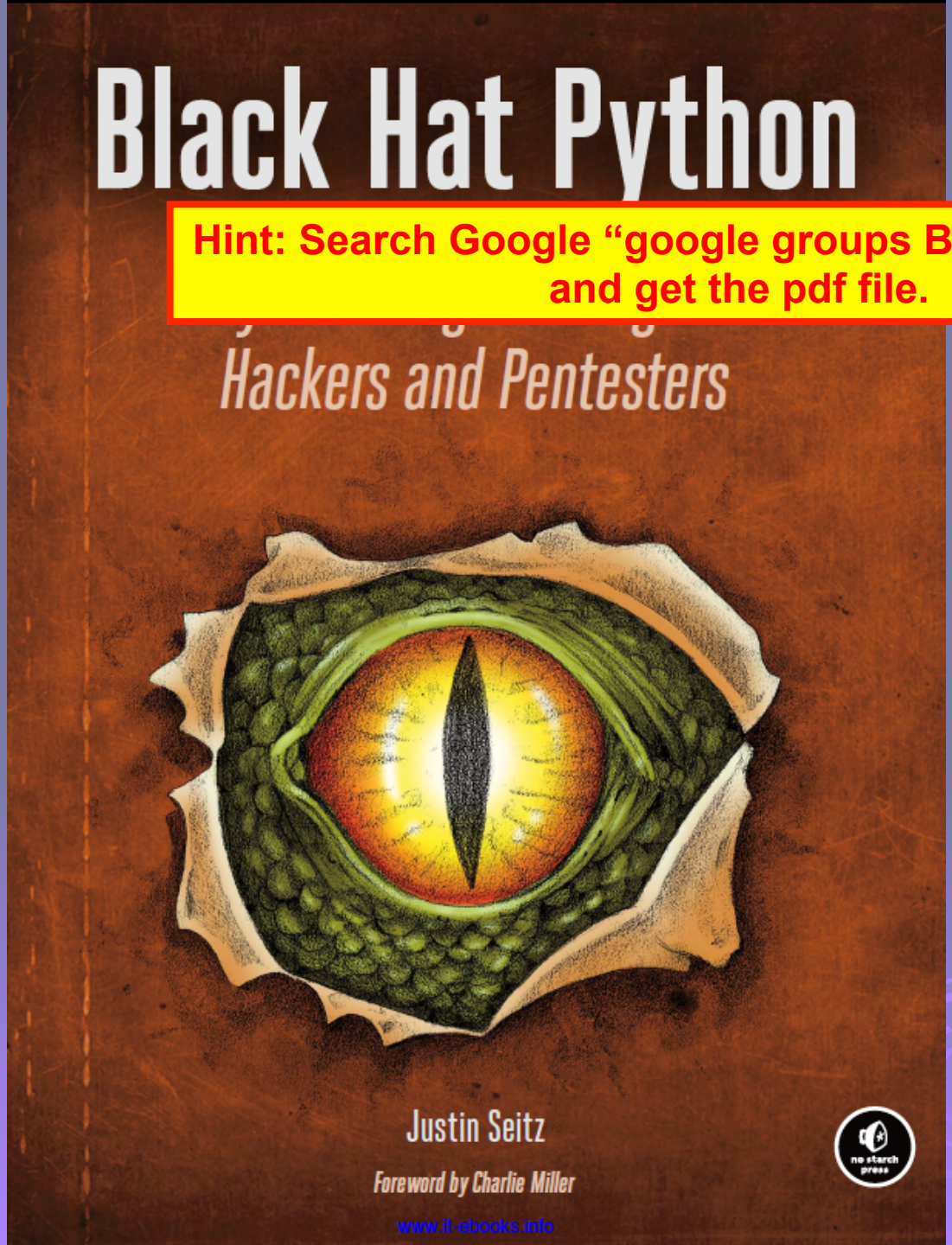
Black Hat Python

Python Programming for Hackers and Pentesters

Justin Seitz

Foreword by Charlie Miller

no starch press



Hint: Search Google "google groups Black Hat Python" and get the pdf file.

Course plan

TTM4536 - Ethical Hacking - Information Security, Specialization Course

Examination arrangement

Examination arrangement: Portfolio assessment
Grade: Letters

Evaluation form	Weighting	Duration	Examination aids
Work	20/100		A
Work	20/100		A
Oral examination	60/100	30 minutes	D

Course content

The course covers the main techniques used by computer hackers and penetration testers in order to better defend against intrusions and security violations in live systems, including low-level kernel and hardware topics, techniques for web applications, exploit techniques, rootkits and some audit techniques used in digital forensics.

Learning outcome

A. Knowledge: Students will learn the underlying principles and techniques associated with the cybersecurity practice known as penetration testing or ethical hacking. They will become familiar with the entire penetration testing process including planning, reconnaissance, scanning, exploitation, post-exploitation and result reporting. B. Skills: For every offensive penetration technique the students will learn the corresponding remedial technique. By this, the students will develop a practical understanding of the current cybersecurity issues and the ways how the errors made by users, administrators, or programmers can lead to exploitable insecurities.

Learning methods and activities

Lectures, seminars, invited lectures, student presentations and laboratory exercises. Two compulsory practical ethical hacking tasks; both tasks must be approved to qualify for the final exam.

Compulsory assignments

Work 1
Work 2

Further on evaluation

Portfolio assessment is the basis for the grade in the course. The portfolio includes two practical ethical hacking tasks which each counts 20% and a oral final exam which counts 60%. The results for the parts are given in %-scores. The entire portfolio is assigned a letter grade. The oral exam is given in English only.
If a student also after the re-sit exam has the final grade F/failed, the student must repeat the entire course. Works that count in the final grade must be repeated.

More on the course

Lecturers page

Facts

Version: 1
Credits: 7.5 SP
Study level: Second degree level

Coursework

Term no.: 1
Teaching semester: AUTUMN 2017

No.of lecture hours: 1
Lab hours: 4
No.of specialization hours: 7

Language of instruction: English

Location: Trondheim

Subject area(s)

IKT

Sivilingeniør

Technological subjects

Telematics

Contact information

Course coordinator:
Danilo Gligoroski

Lecturer(s):
Danilo Gligoroski

Department with academic responsibility
Department of Information Security and Communication Technology

Phone: 73 59 43 24

Course plan

TTM4536 - Ethical Hacking - Information Security, Specialization Course

About | **Timetable** | Examination

Autumn 2017/Spring 2018

Detailed timetable

Alle

2017 HØST

Day	Time	Weeks	Type	Planned for	Room
Tuesday	12:15 - 13:00	34-47	Forelesning/Lab	MSSECMOB, MSTCNNS, MTKOM	EL4
Tuesday	13:15 - 17:00	34-47	Forelesning/Lab		

Course plan

TTM4536 - E

About | Timet

These classes, will introduce the content of the material in a form of slides, videos and instructions.

Detailed timetable

Alle

2017 HØST

Day	Time	Weeks	Type	Planned for	Room
Tuesday	12:15 - 13:00	34-47	Forelesning/Lab	MSSECMOB, MSTCNNS, MTKOM	EL4
Tuesday	13:15 - 17:00	34-47	Forelesning/Lab		

Course plan

TTM4536 - E

About | Timet

These classes, will introduce the content of the material in a form of slides, videos and instructions.

Detailed timetable

Alle

2017 HØST

Day	Time	Weeks	Type	Planned for	Room
Tuesday	12:15 - 13:00	34-47	Forelesning/Lab	MSSECMOB, MSTCNNS, MTKOM	EL4
Tuesday	13:15 - 17:00	34-47	Forelesning/Lab		

These laboratory classes will be in Sahara lab. I will be present at least in the first 60 minutes to see how the students are repeating the exercises given in the slides. I will have one student teaching assistant that will be available for consultations during the lab time.

Course plan

Course plan

From the textbook we will cover at least 8 chapters

Course plan

From the textbook we will cover at least 8 chapters

We will install Virtual box images of Kali Linux and Windows 7. The best strategy is to install them on your laptop. It will be also possible to use the machines in Sahara laboratory.

All experiments will be on those virtual machines.

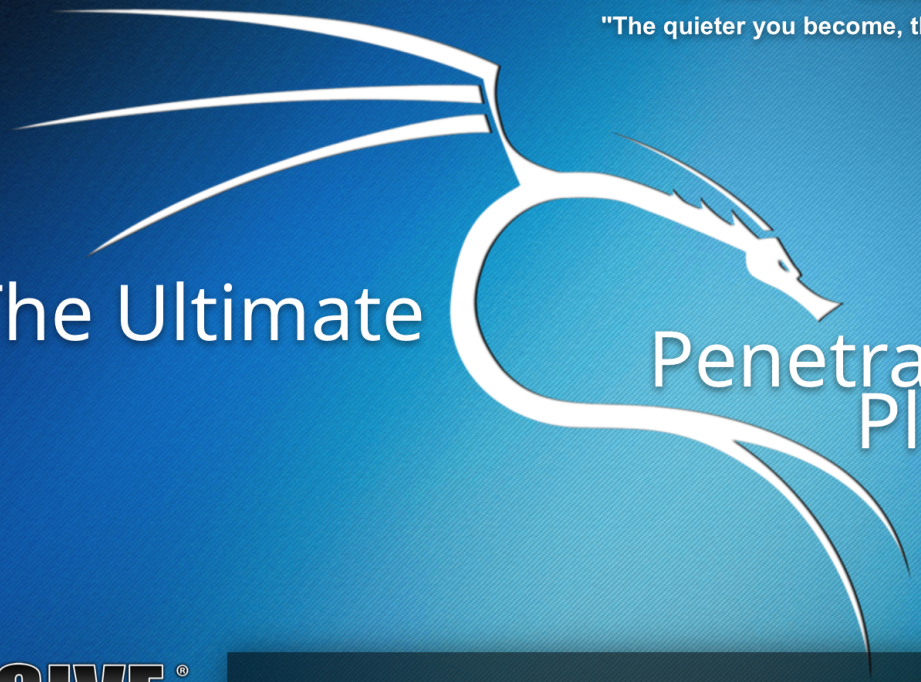
KALI LINUX™

"the quieter you become, the more you are able to hear"

Blog Downloads Training Documentation Community About Us

Kali Linux 2.0

"The quieter you become, the more you are able to hear."



The Ultimate

Penetration Testing Platform

OFFENSIVE® security

From the creators of BackTrack comes Kali Linux, the most **advanced and versatile penetration testing platform** ever created. We have a set of amazing features lined up in our security distribution geared at **streamlining the penetration testing experience**.

Course plan

CTF TIME

Home / CTF? WTF?

We will have 3-4 weeks devoted to solving (analyzing) CTF problems. The problems will be mostly Binary exploits, Web exploits and Crypto problems.

CTF? WTF?

Capture the Flag (CTF) is a special kind of information security competitions. There are three common types of CTFs: Jeopardy, Attack-Defence and mixed.

Jeopardy-style CTFs has a couple of questions (tasks) in range of categories. For example, Web, Forensic, Crypto, Binary or something else. Team can gain some points for every solved task. More points for more complicated tasks usually. The next task in chain can be opened only after some team solve previous task. Then the game time is over sum of points shows you a CTF winer. Famous example of such CTF is Defcon CTF quals.

Well, **attack-defence** is another interesting kind of competitions. Here every team has own network(or only one host) with vulnarable services. Your team has time for patching your services and developing exploits usually. So, then organizers connects participants of competition and the wargame starts! You should protect own services for defence points and hack opponents for attack points. Historically this is a first type of CTFs, everybody knows about DEF CON CTF - something like a World Cup of all other competitions.

Mixed competitions may vary possible formats. It may be something like wargame with special time for task-based elements (like UCSB iCTF).

CTF games often touch on many other aspects of information security: **cryptography, stego, binary analysis, reverse engeneering, mobile security** and others. Good teams generally have strong skills and experience in all these issues.

Course plan

TTM4536 - Ethical Hacking - Information Security, Specialization Course

About | Timetable | **Examination**

Autumn 2017/Spring 2018

Examination arrangement: Portfolio assessment

Term	Statuskode	Evaluation form	Weighting	Examination aids	Date	Time	Room *
Autumn	ORD	Work	20/100	A			
Autumn	ORD	Work	20/100	A			
Autumn	ORD	Oral examination	60/100	D			

* The location (room) for a written examination is published 3 days before examination date.

If more than one room is listed, you will find your room at Studentweb.

Examination

For more information regarding registration for examination and examination procedures, see "Innsida - Exams"

More on examinations at NTNU

Course plan

A presentation of a step-by-step solution of one easy CTF problem. I will assign those problem to students during the first two weeks of the course.

TTM4536 - Ethical Hacking - Information Security, Specialization Course

About | Timetable | **Examination** | Autumn 2017/Spring 2018

Examination arrangement: Portfolio assessment

Term	Statuskode	Evaluation form	Weighting	Examination aids	Date	Time	Room *
Autumn	ORD	Work	20/100	A			
Autumn	ORD	Work	20/100	A			
Autumn	ORD	Oral examination	60/100	D			

* The location (room) for a written examination is published 3 days before examination date.

If more than one room is listed, you will find your room at Studentweb.

Examination

For more information regarding registration for examination and examination procedures, see "Innsida - Exams"

More on examinations at NTNU

Course plan

A presentation of a step-by-step solution of one easy CTF problem. I will assign those problem to students during the first two weeks of the course.

TTM4536 - Ethical Hacking - Information Security, Specialization Course

About | Timetable | **Examination** | Autumn 2017/Spring 2018

Examination arrangement: Portfolio assessment

Term	Statuskode	Evaluation form	Weighting	Examination aids	Date	Time	Room *
Autumn	ORD	Work	20/100	A			
Autumn	ORD	Work	20/100	A			
Autumn	ORD	Oral examination	60/100	D			

* The location (room) for a written examination is published 3 days before examination date.

If more than one room is listed, you will find your room at Studentweb.

Examination

For more information regarding registration for examination and examination procedures, see "Innsida - Exams"

More on examinations at NTNU

A presentation of a step-by-step solution of another CTF "Crypto" or "Webapp exploit" or "Forensics" problem. I will assign those problem to students during the second week of October 2016.

Course plan

A presentation of a step-by-step solution of one easy CTF problem. I will assign those problem to students during the first two weeks of the course.

TTM4536 - Ethical Hacking - Information Security, Specialization Course

About | Timetable | **Examination** | Autumn 2017/Spring 2018

Examination arrangement: Portfolio assessment

Term	Statuskode	Evaluation form	Weighting	Examination aids	Date	Time	Room *
Autumn	ORD	Work	20/100	A			
Autumn	ORD	Work	20/100	A			
Autumn	ORD	Oral examination	60/100	D			

* The location (room) for a written examination is published 3 days before examination date.

If more than one room is listed, you will find ...

Examination

For more information regarding registration for examination and examination procedures, see "Innsida - Exams"

More on examinations at NTNU

Questions will be from the textbook material and from the slides and instructions posted on Blackboard.

A presentation of a step-by-step solution of another CTF "Crypto" or "Webapp exploit" or "Forensics" problem. I will assign those problem to students during the second week of October 2016.

Course plan

Questions for the final exam will be from the textbook material, and from the slides and instructions posted on Blackboard.

TTM

Abou

Exam

Te							
Autumn	ORD	Work	20/100	A			
Autumn	ORD	Work	20/100	A			
Autumn	ORD	Oral examination	60/100	D			

ling and examination procedures, see "Innsida - Exams"

More on examinations at NTNU

* The location (room) for a written examination is published 3 days before examination date.

If more than one room is listed, you will find your room at Studentweb.

TTM4536 - Ethical Hacking - Information Security, Specialization Course

- Plan for 29 Aug 2017

Following the instruction of Chapter 1 from the textbook

- Install Kali Linux from the scratch
  - You can choose to install it on your laptop
  - Or you can install it on USB stick (not recommended – last year experiences)

Following the instruction of Chapter 1 from the textbook

- Once the Kali Linux is installed log in as root and install VirtualBox Guest Additions
  - See the instructions in the pdf file submitted on Blackboard

Following the instruction of Chapter 1 from the textbook

- Once the Kali Linux is installed log in as root and install VirtualBox Guest Additions
  - See the instructions in the pdf file submitted on Blackboard
- Check the installed Python version
  - (for this course it should be Python 2.7)

Following the instruction of Chapter 1 from the textbook

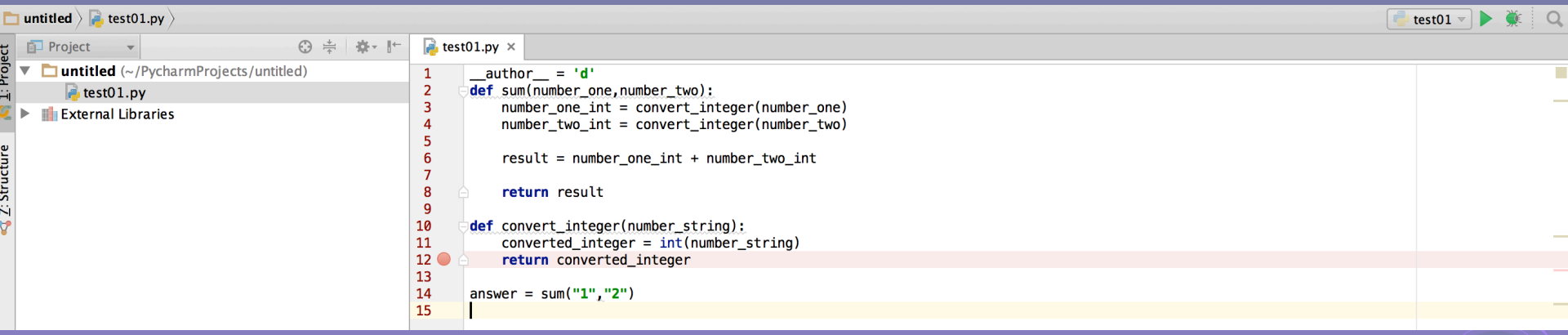
- Once the Kali Linux is installed log in as root and install VirtualBox Guest Additions
  - See the instructions in the pdf file submitted on Blackboard
- Check the installed Python version
  - (for this course it should be Python 2.7)
- Install a Python IDE
  - A good commercial one is WingIDE
  - A good open source one is PyCharm

Hint

- If you face problems with installing or running Kali Linux today, the following exercises with PyCharm can be performed on the Linux machines in Sahara lab.

Becoming familiar with PyCharm

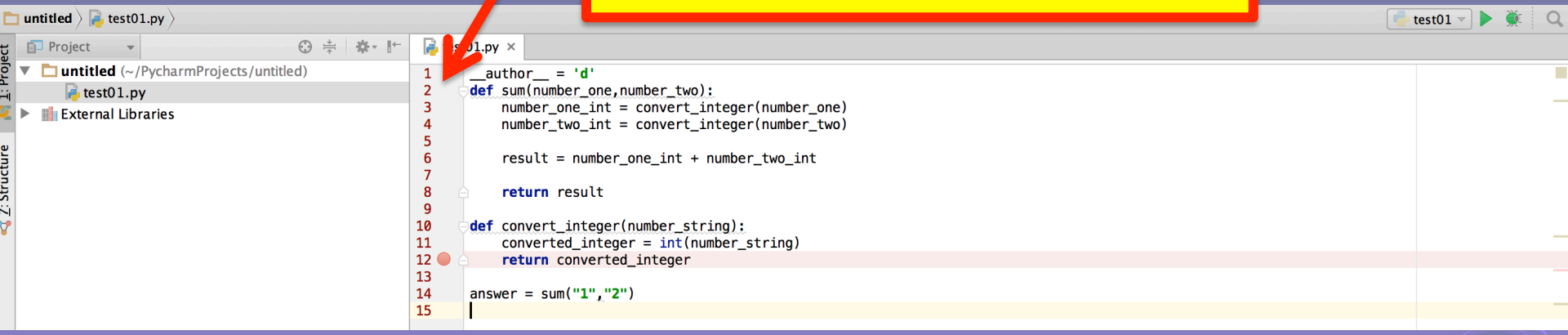
- Start a new project in PyCharm
- Type the simple program test01.py



Becoming familiar with PyCharm

- Start a new pr
- Type the simp

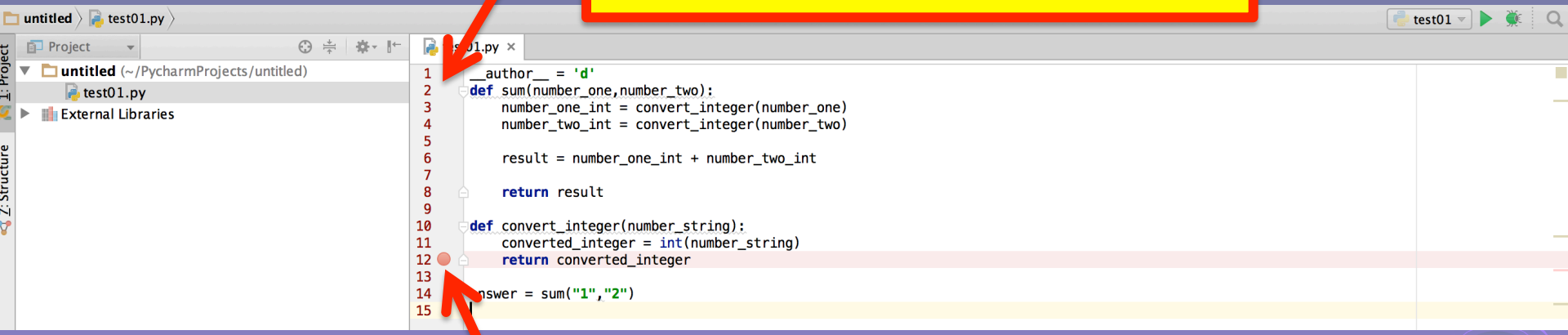
Right click with the mouse and choose to show line numbers



Becoming familiar with PyCharm

- Start a new pr
- Type the simp

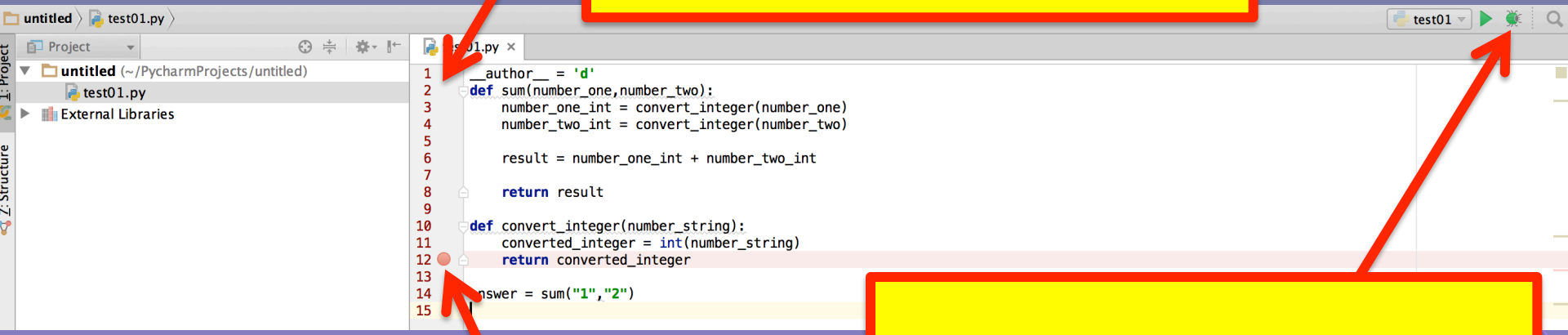
Right click with the mouse and choose to show line numbers



Set a Break Point

Becoming familiar with PyCharm

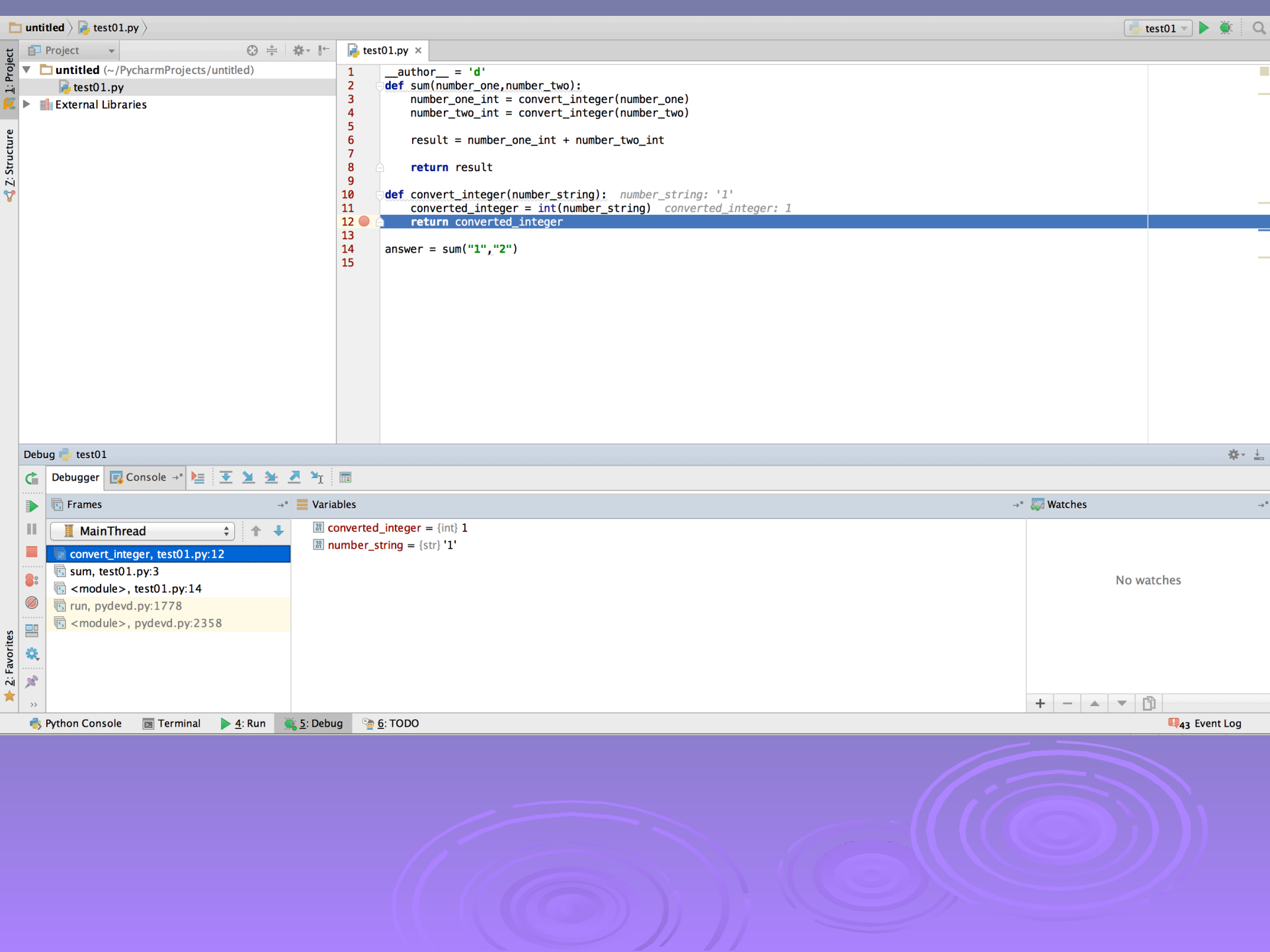
- Start a new pr
- Type the simp

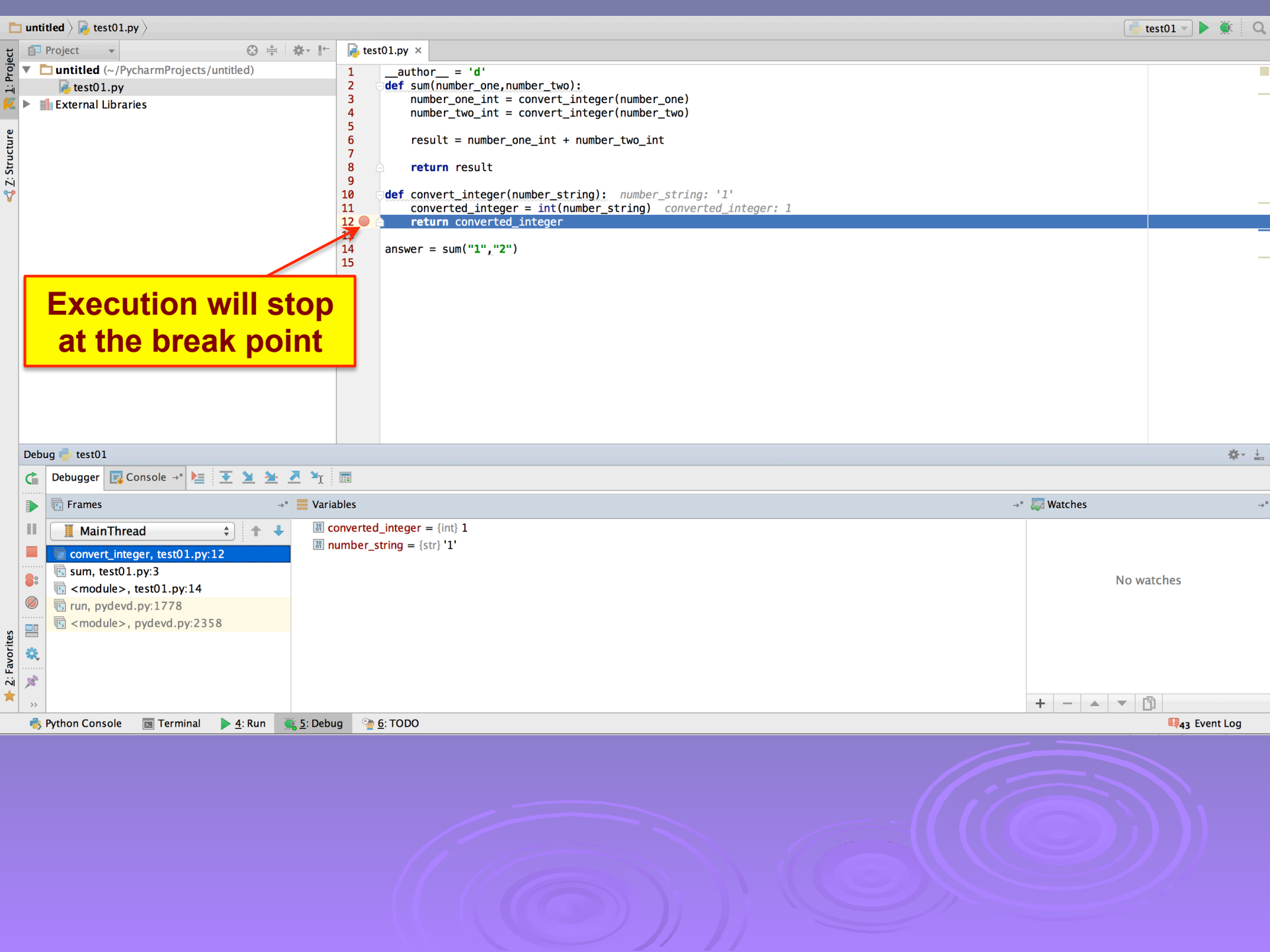


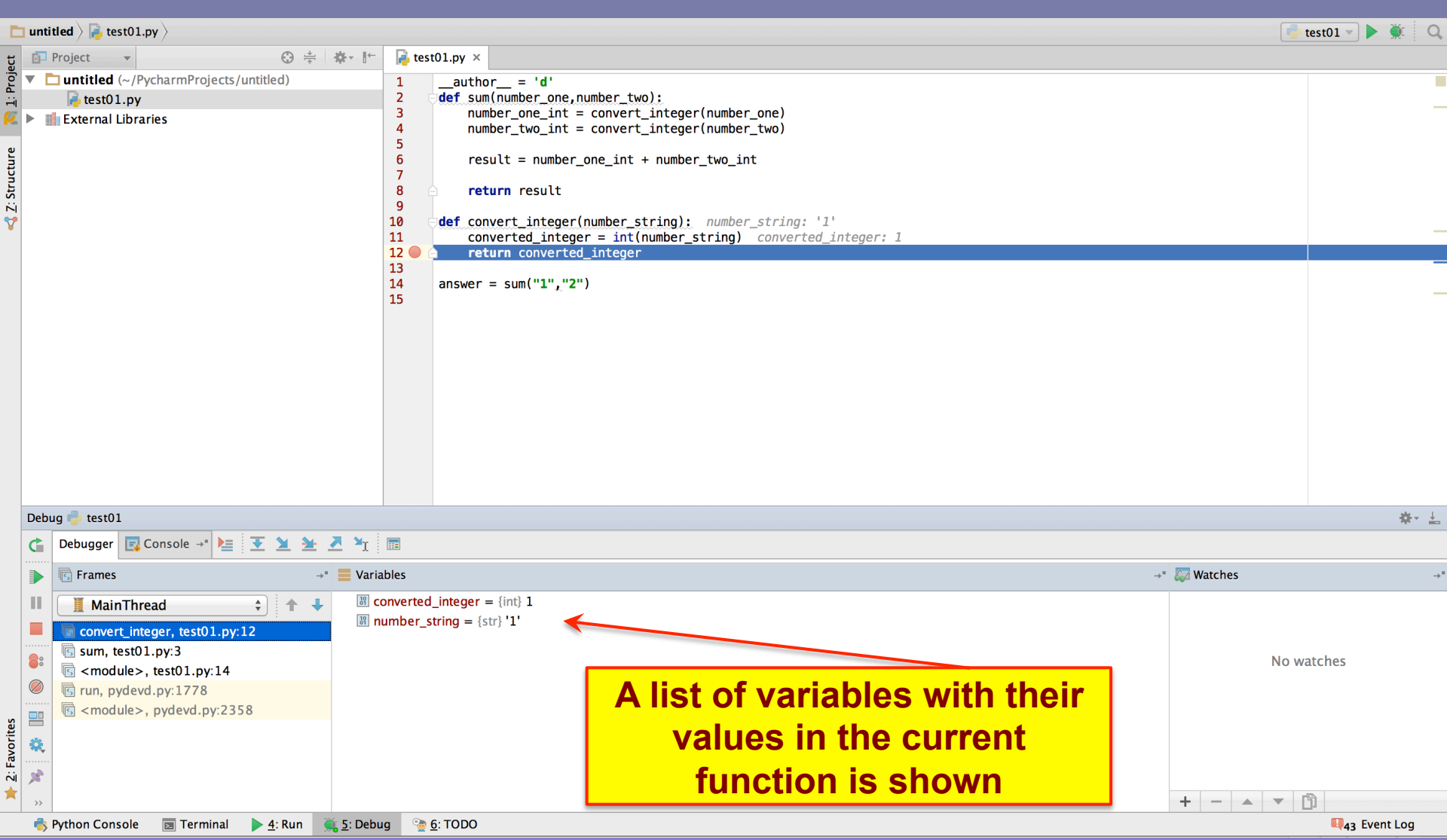
Right click with the mouse and choose to show line numbers

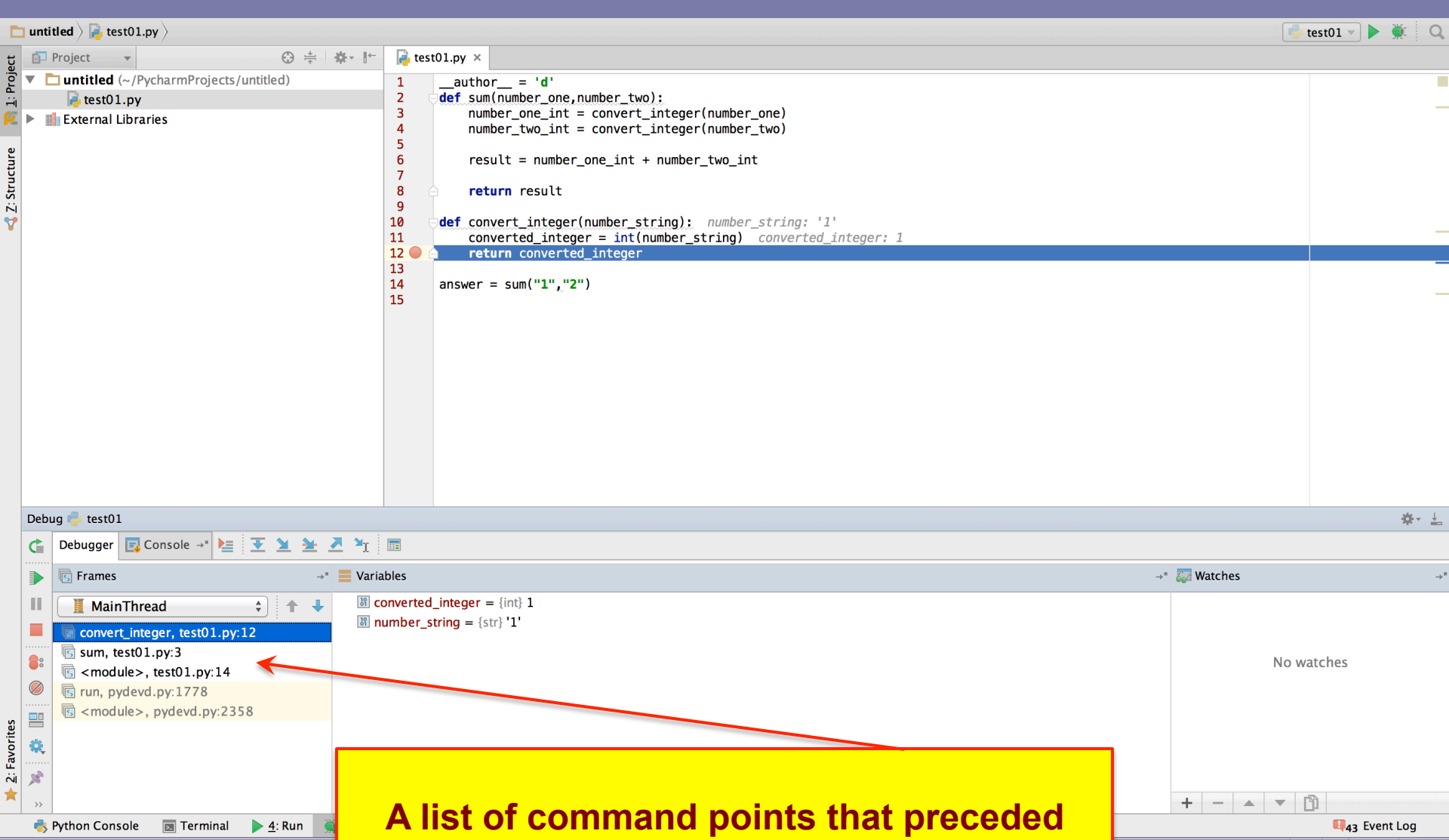
Choose to start a Debugging session

Set a Break Point

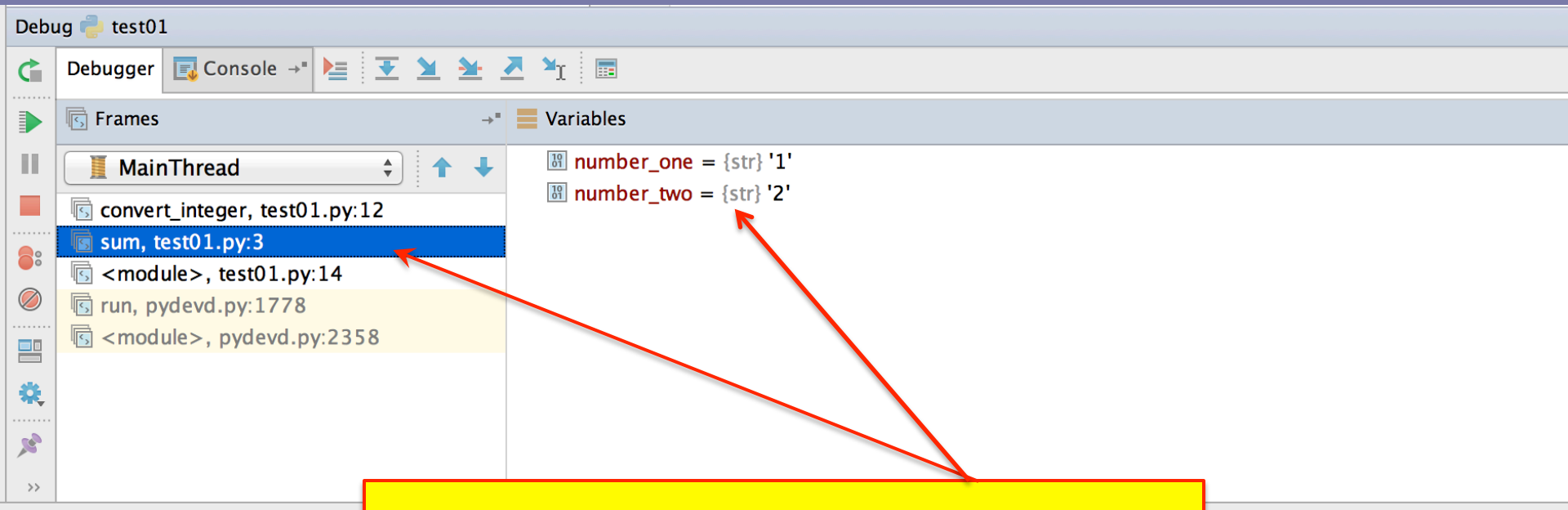

untitled
test01.py
test01
Project
untitled (~/PycharmProjects/untitled)
test01.py
External Libraries
test01.py
1 __author__ = 'd'
2 def sum(number_one,number_two):
3 number_one_int = convert_integer(number_one)
4 number_two_int = convert_integer(number_two)
5
6 result = number_one_int + number_two_int
7
8 return result
9
10 def convert_integer(number_string): number_string: '1'
11 converted_integer = int(number_string) converted_integer: 1
12 return converted_integer
13
14 answer = sum("1","2")
15
Debug test01
Debugger
Console
Frames
Variables
Watches
MainThread
convert_integer, test01.py:12
sum, test01.py:3
<module>, test01.py:14
run, pydevd.py:1778
<module>, pydevd.py:2358
converted_integer = {int} 1
number_string = {str} '1'
No watches
Python Console
Terminal
4: Run
5: Debug
6: TODO
Event Log
1: Project
7: Structure
2: Favorites


untitled
test01.py
test01
Project
untitled (~/PycharmProjects/untitled)
test01.py
External Libraries
test01.py
1 __author__ = 'd'
2 def sum(number_one,number_two):
3 number_one_int = convert_integer(number_one)
4 number_two_int = convert_integer(number_two)
5
6 result = number_one_int + number_two_int
7
8 return result
9
10 def convert_integer(number_string): number_string: '1'
11 converted_integer = int(number_string) converted_integer: 1
12 return converted_integer
13
14 answer = sum("1","2")
15
Execution will stop at the break point
Debug test01
Debugger
Console
Frames
Variables
Watches
MainThread
convert_integer, test01.py:12
sum, test01.py:3
<module>, test01.py:14
run, pydevd.py:1778
<module>, pydevd.py:2358
converted_integer = {int} 1
number_string = {str} '1'
No watches
Python Console
Terminal
4: Run
5: Debug
6: TODO
Event Log
1: Project
7: Structure
2: Favorites

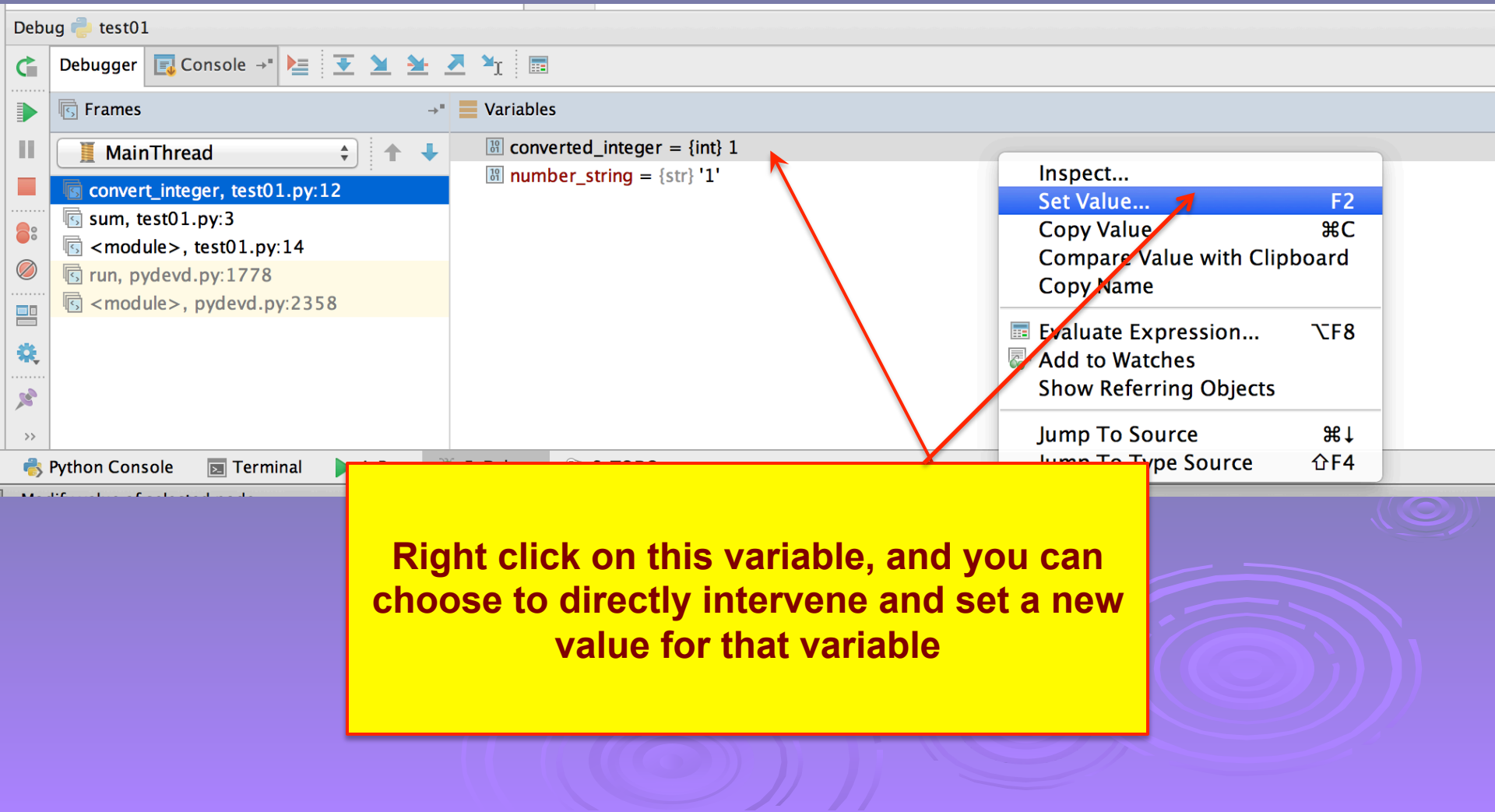

untitled
test01.py
test01
Project
untitled (~/PycharmProjects/untitled)
test01.py
External Libraries
test01.py
__author__ = 'd'
def sum(number_one,number_two):
number_one_int = convert_integer(number_one)
number_two_int = convert_integer(number_two)
result = number_one_int + number_two_int
return result
def convert_integer(number_string): number_string: '1'
converted_integer = int(number_string) converted_integer: 1
return converted_integer
answer = sum("1","2")
Debug test01
Debugger
Console
Frames
Variables
Watches
MainThread
convert_integer, test01.py:12
sum, test01.py:3
<module>, test01.py:14
run, pydevd.py:1778
<module>, pydevd.py:2358
converted_integer = {int} 1
number_string = {str} '1'
No watches
A list of variables with their values in the current function is shown
Python Console
Terminal
4: Run
5: Debug
6: TODO
Event Log
1: Project
7: Structure
2: Favorites



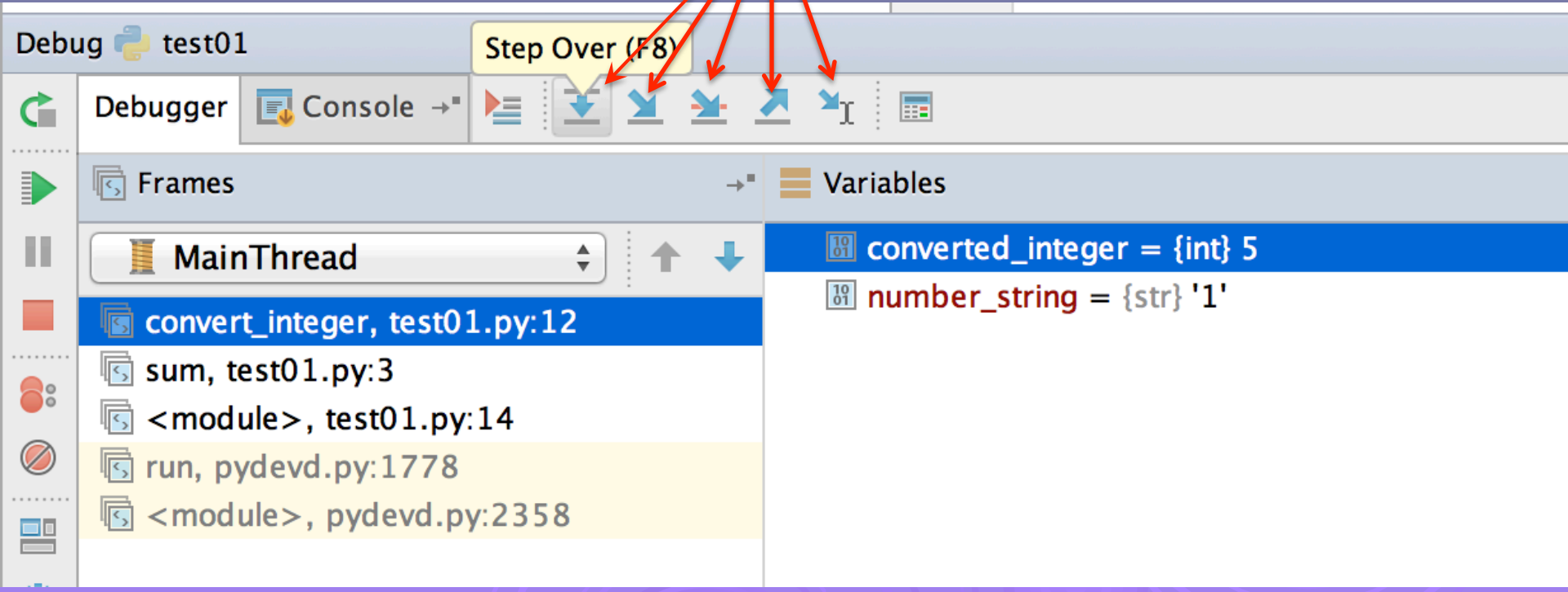
A list of command points that preceded this current state is shown. In this example: the function sum was called, which called the function convert_integer

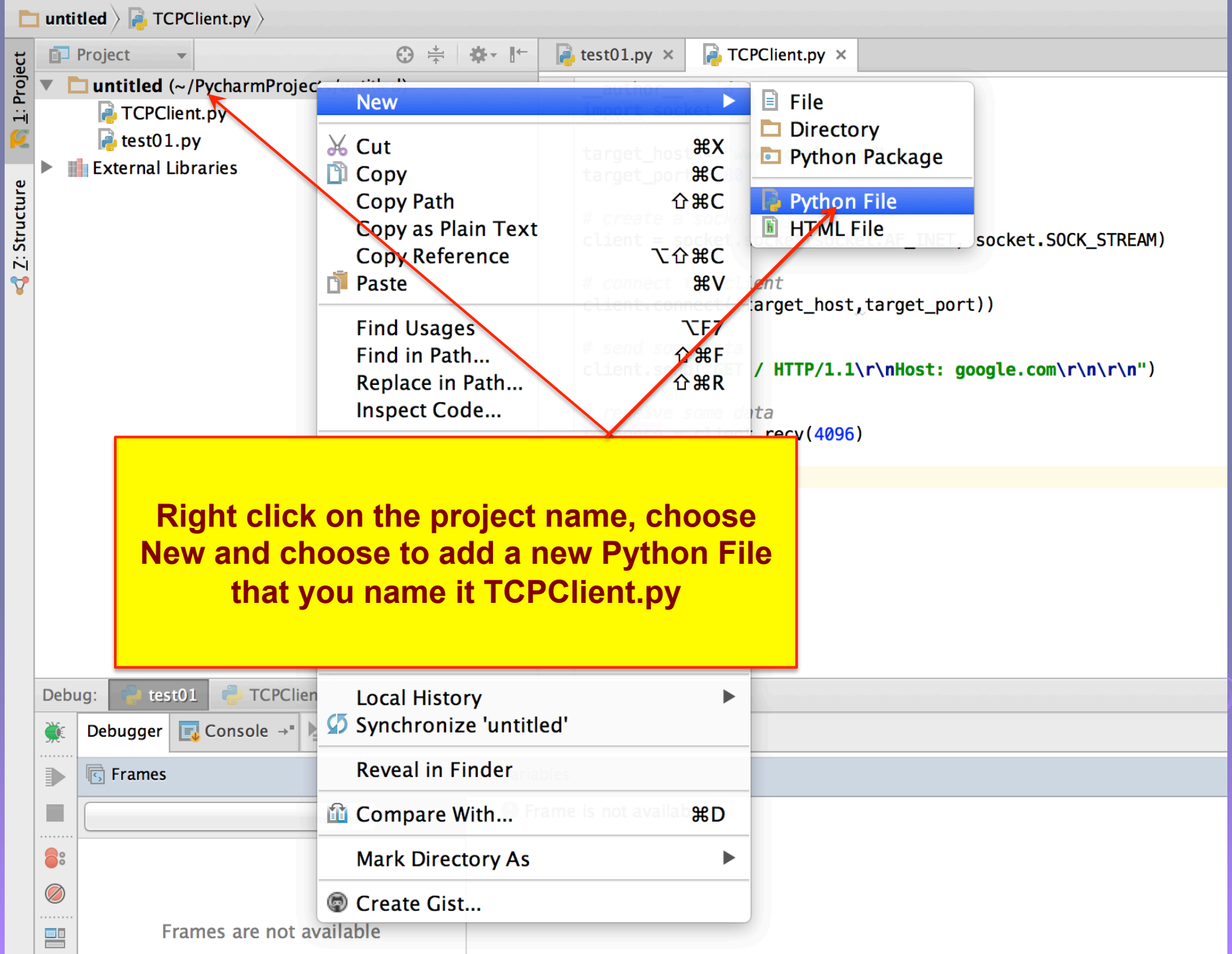


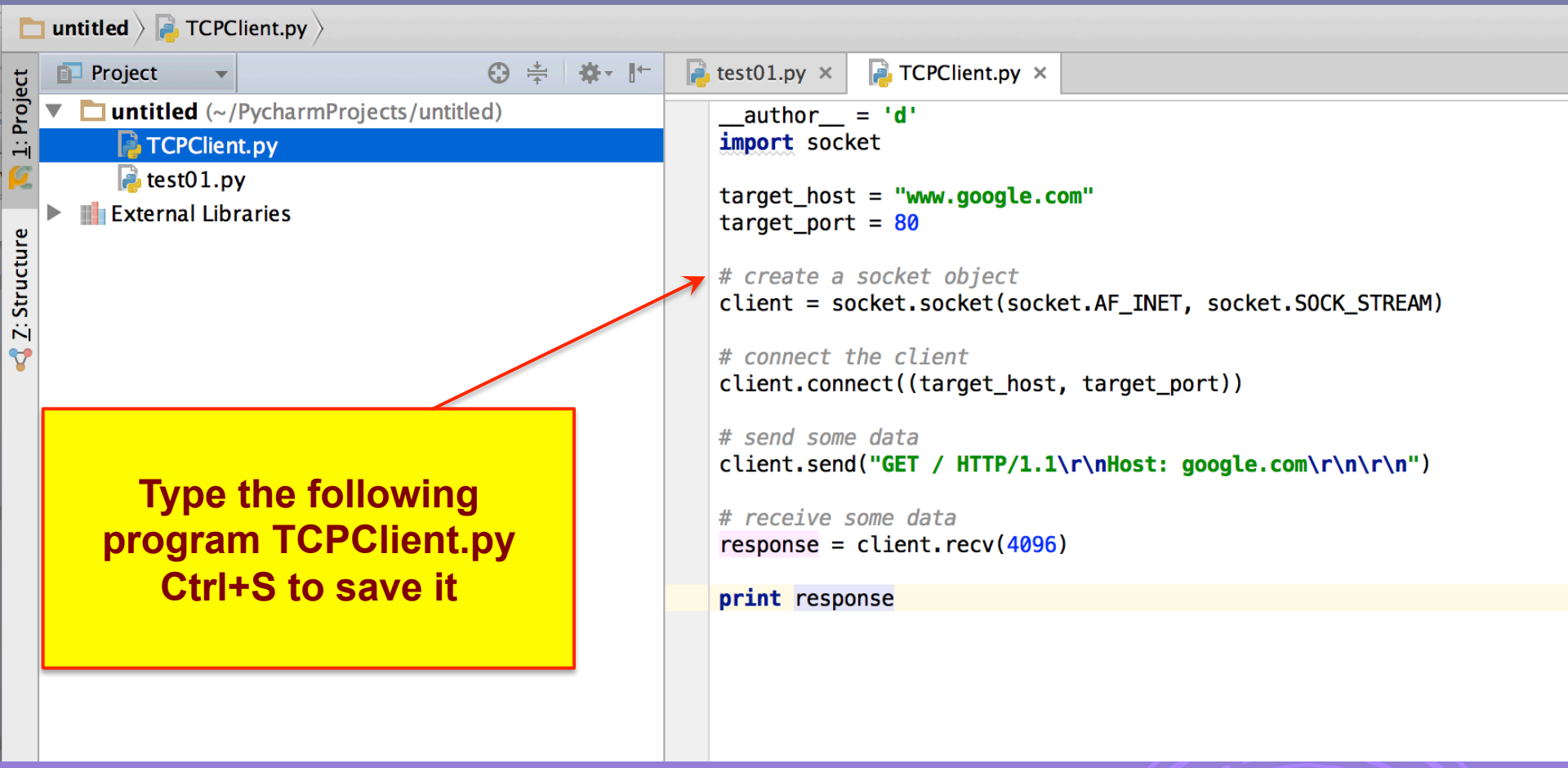
If you click on the function name that was called initially, you will see the list of the arguments with their values

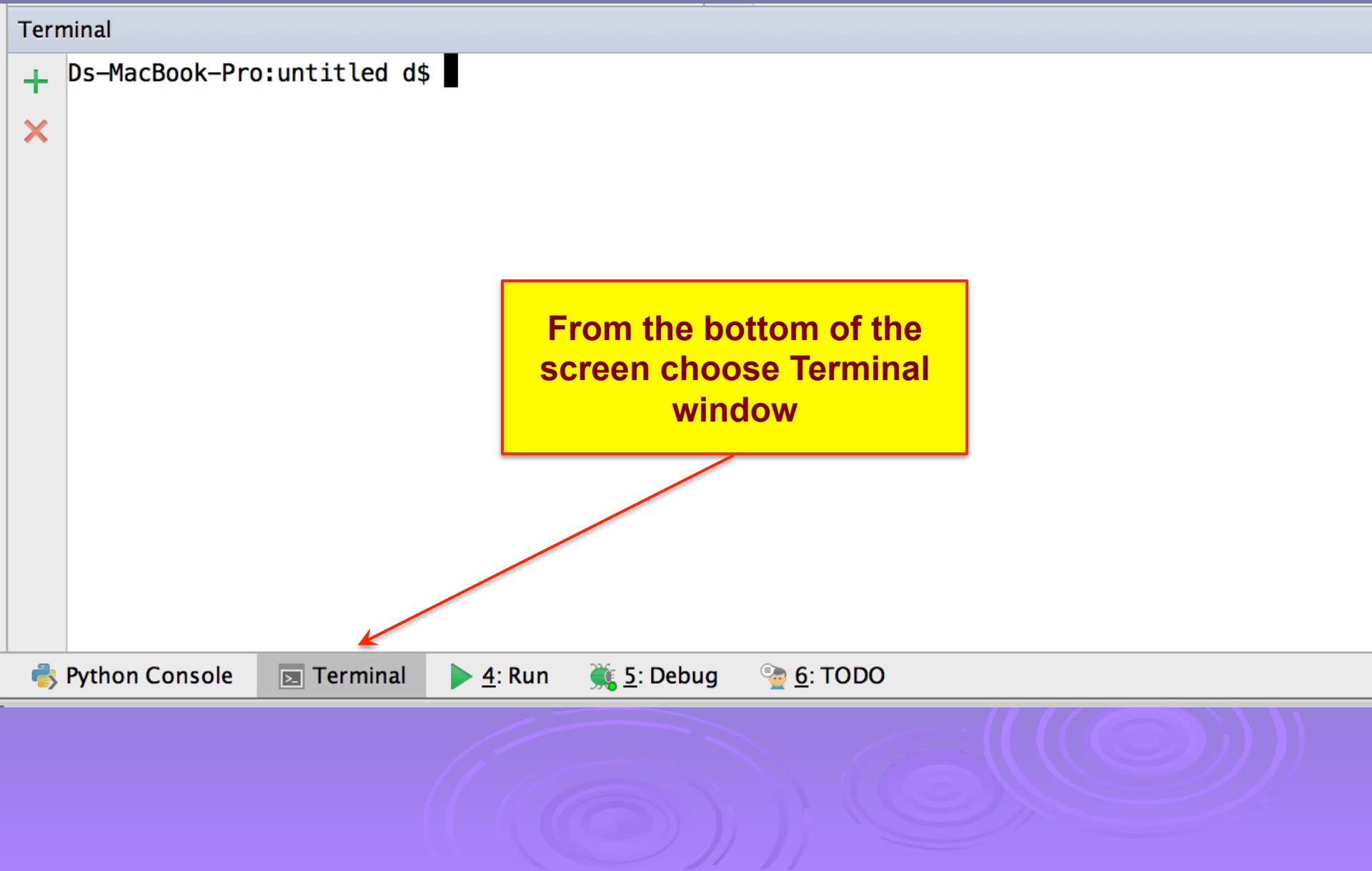

Debug test01
Debugger
Console
Frames
Variables
MainThread
convert_integer, test01.py:12
sum, test01.py:3
<module>, test01.py:14
run, pydevd.py:1778
<module>, pydevd.py:2358
converted_integer = {int} 1
number_string = {str} '1'
Inspect...
Set Value... F2
Copy Value ⌘C
Compare Value with Clipboard
Copy Name
Evaluate Expression... ⌥F8
Add to Watches
Show Referring Objects
Jump To Source ⌘↓
Jump To Type Source ⇧F4
Python Console
Terminal
Right click on this variable, and you can choose to directly intervene and set a new value for that variable

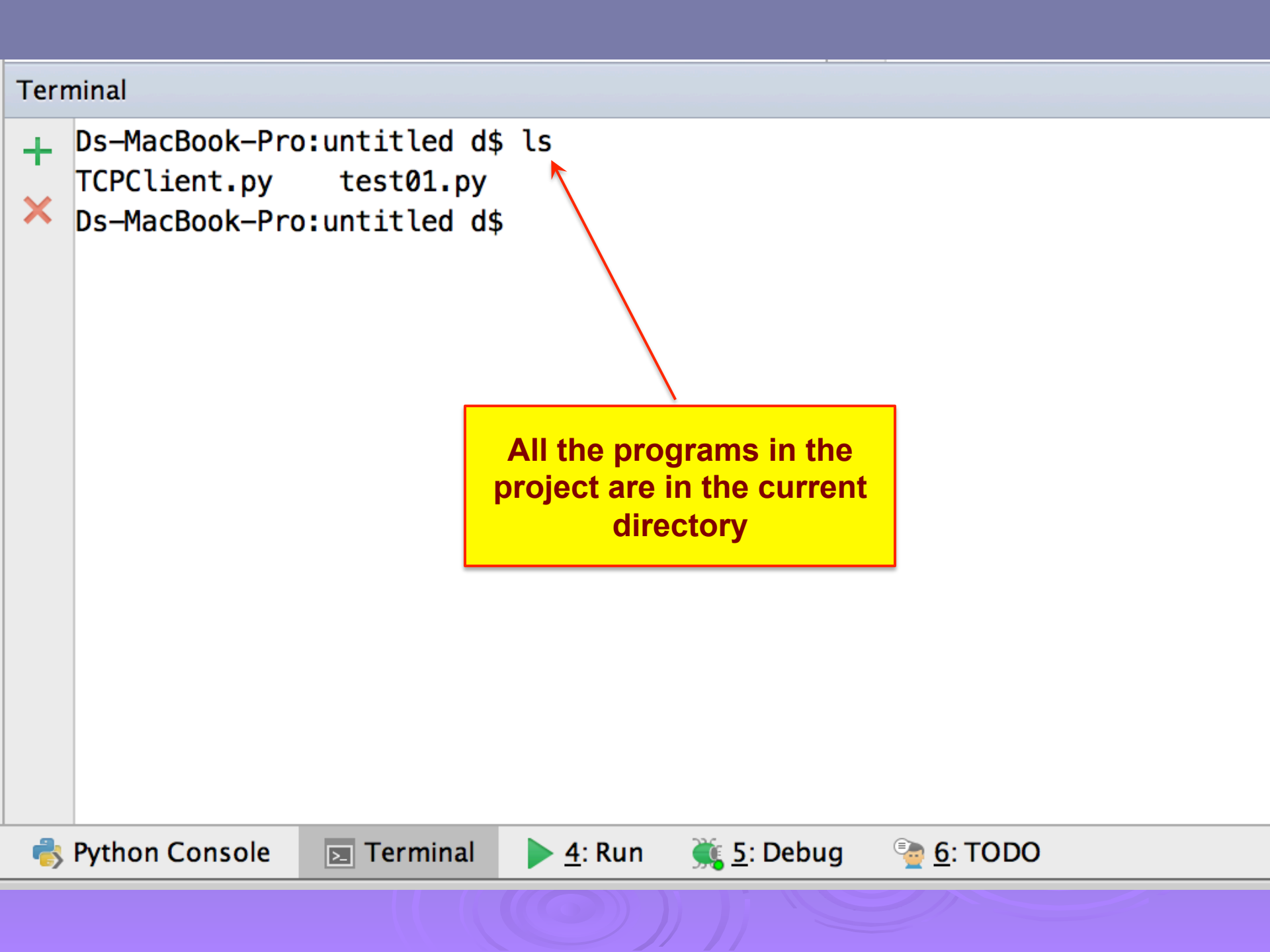
Control the execution of the program, step by step (Step Over, Step Into, Step Into My Code, Step Out, Run To Cursor)

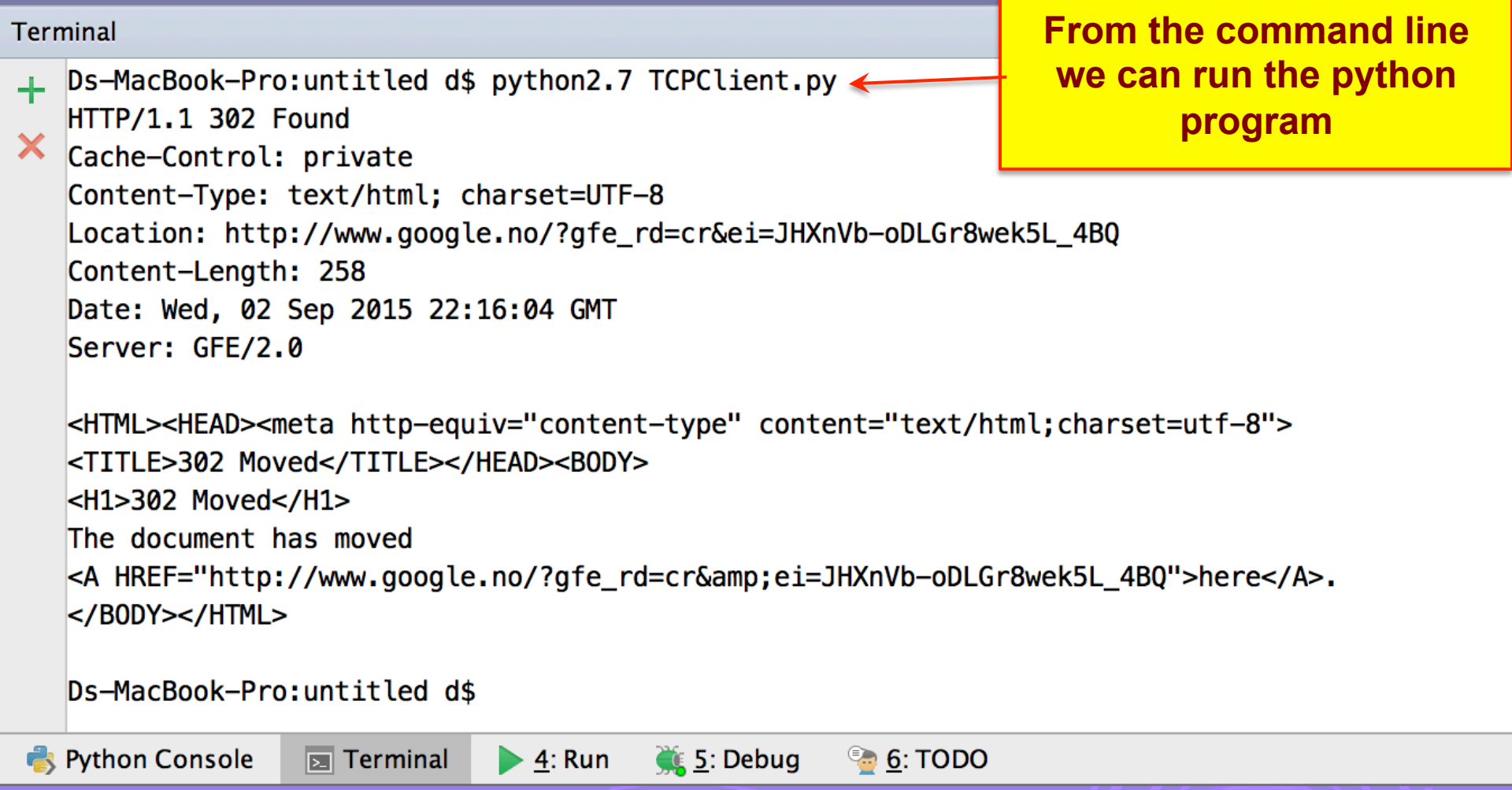



untitled
TCPClient.py
Project
test01.py
TCPClient.py
untitled (~/PycharmProject
TCPClient.py
test01.py
External Libraries
New
Cut ⌘X
Copy ⌘C
Copy Path ⇧⌘C
Copy as Plain Text
Copy Reference ⌥⇧⌘C
Paste ⌘V
Find Usages ⌥F7
Find in Path... ⇧⌘F
Replace in Path... ⇧⌘R
Inspect Code...
File
Directory
Python Package
Python File
HTML File
socket.SOCK_STREAM)
target_host,target_port))
/ HTTP/1.1\r\nHost: google.com\r\n\r\n")
recv(4096)
Right click on the project name, choose New and choose to add a new Python File that you name it TCPClient.py
Local History
Synchronize 'untitled'
Reveal in Finder
Compare With... ⌘D
Mark Directory As
Create Gist...
Debug:
test01
Debugger
Console
Frames
Frames are not available
1: Project
Z: Structure


untitled
TCPClient.py
Project
untitled (~/PycharmProjects/untitled)
TCPClient.py
test01.py
External Libraries
1: Project
7: Structure
test01.py
TCPClient.py
__author__ = 'd'
import socket
target_host = "www.google.com"
target_port = 80
# create a socket object
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# connect the client
client.connect((target_host, target_port))
# send some data
client.send("GET / HTTP/1.1\r\nHost: google.com\r\n\r\n")
# receive some data
response = client.recv(4096)
print response
Type the following program TCPClient.py Ctrl+S to save it


Terminal
Ds-MacBook-Pro:untitled d$
From the bottom of the screen choose Terminal window
Python Console
Terminal
4: Run
5: Debug
6: TODO


Terminal
Ds-MacBook-Pro:untitled d$ ls
TCPClient.py test01.py
Ds-MacBook-Pro:untitled d$
All the programs in the project are in the current directory
Python Console
Terminal
4: Run
5: Debug
6: TODO


Terminal
Ds-MacBook-Pro:untitled d$ python2.7 TCPClient.py
HTTP/1.1 302 Found
Cache-Control: private
Content-Type: text/html; charset=UTF-8
Location: http://www.google.no/?gfe_rd=cr&ei=JHXnVb-oDLGr8wek5L_4BQ
Content-Length: 258
Date: Wed, 02 Sep 2015 22:16:04 GMT
Server: GFE/2.0
<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">
<TITLE>302 Moved</TITLE></HEAD><BODY>
<H1>302 Moved</H1>
The document has moved
<A HREF="http://www.google.no/?gfe_rd=cr&ei=JHXnVb-oDLGr8wek5L_4BQ">here</A>.
</BODY></HTML>
Ds-MacBook-Pro:untitled d$
From the command line we can run the python program
Python Console
Terminal
4: Run
5: Debug
6: TODO

TCPServer.py

```
import socket
import threading

bind_ip = "0.0.0.0"
bind_port = 9999

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

server.bind((bind_ip,bind_port))

server.listen(5)

print "[*] Listening on %s:%d" % (bind_ip,bind_port)

# this is our client-handling thread

def handle_client(client_socket):

    # print out what the client sends
    request = client_socket.recv(1024)

    print "[*] Received: %s" % request

    # send back a packet
    client_socket.send("ACK!")

    client_socket.close()

while True:
    client,addr = server.accept()

    print "[*] Accepted connection from: %s:%d" % (addr[0],addr[1])

    # spin up our client thread to handle incoming data
    client_handler = threading.Thread(target=handle_client,args=(client,))
    client_handler.start()
```

TCPClient02.py

```
import socket

target_host = "0.0.0.0"
target_port = 9999

# create a socket object
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# connect the client
client.connect((target_host, target_port))

# send some data
client.send("ABCDEF")

# receive some data
response = client.recv(4096)

print response
```

- In the Terminal start the TCPServer with
- `python2.7 TCPServer.py &`
- Then if you run TCPClient02.py
- `python2.7 TCPClient02.py`
- You get the following response:

```
[*] Accepted connection from: 127.0.0.1:56356
[*] Received: ABCDEF
ACK!
```

Type in the rest of the programs from Chapter 2 and follow the instructions from the textbook and my oral instructions