

# Course Introduction

BBM 101 - Introduction to Programming I

Hacettepe University

Fall 2016

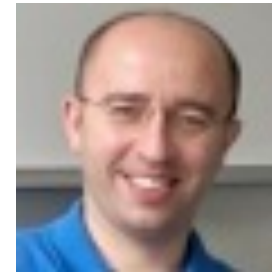
Fuat Akal, Aykut Erdem, Erkut Erdem

# Welcome to BBM 101

- This course teaches **core programming concepts** with an emphasis on **data manipulation tasks** from science, engineering, and business
- **Goal** by the end of the semester: Given a **data source** and a **problem description**, you can independently write a complete, useful program to **solve the problem**

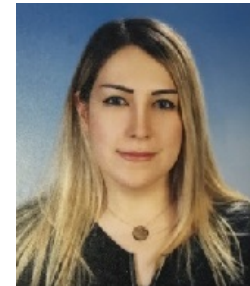
# Course Staff

- Lecturers:
  - Asst. Prof. Dr. Fuat Akal
  - Asst. Prof. Dr. Aykut Erdem
  - Asst. Prof. Dr. Erkut Erdem



# Course Staff

- TAs (Teaching Assistants):
  - Necva Bölücü
  - Selma Dilek
  - Burcu Yalçiner
  - Selim Yılmaz



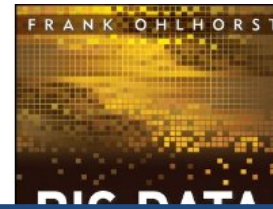
Do not hesitate to ask  
TAs for help!

# Learning Objectives

- Computational problem-solving
  - Writing a program will become your “go-to” solution for data analysis tasks.
- Basic Python proficiency
  - Including experience with relevant libraries for data manipulation, scientific computing, and visualization.
- Experience working with real datasets
  - astronomy, biology, linguistics, oceanography, open government, social networks, and more.
  - You will see that these are easy to process with a program, and that doing so yields insight.

# What This Course is not

- A “skills course” in Python
  - ...though you will become proficient in the basics of the Python programming language
  - ...and you will gain experience with some important Python libraries
- A data analysis / “data science” / data visualization course
  - There will be very little statistics knowledge assumed or taught
- A “project” course
  - the assignments are “real,” but are intended to teach specific programming concepts
- A “software engineering” course
  - Programming is the starting point of computer science and software engineering



*“It’s a great time to be a data geek.”*  
-- Roger Barga, Microsoft Research



*“The greatest minds of my generation are trying to figure out how to make people click on ads”*  
-- Jeff Hammerbacher, co-founder, Cloudera



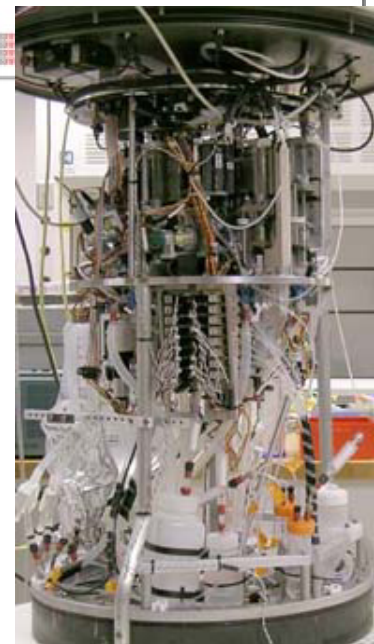
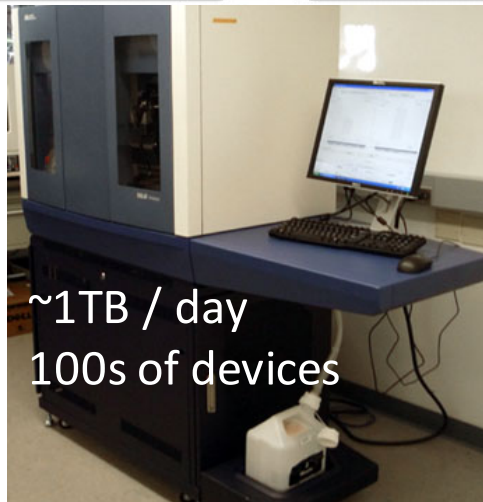
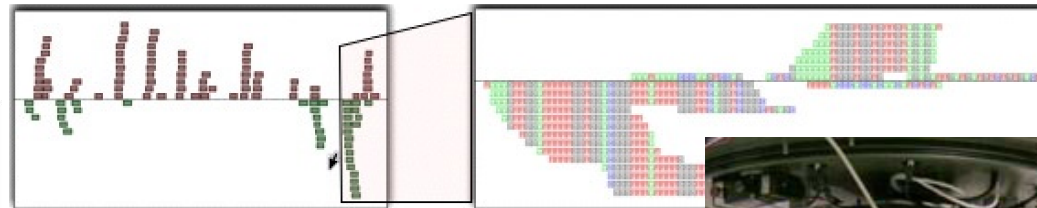
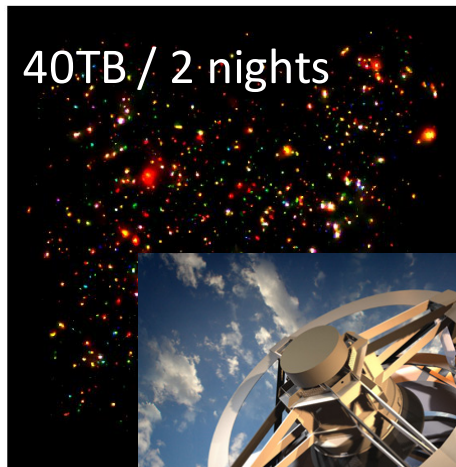


# All of Science is Reducing to Computational Data Manipulation

*Old model: "Query the world" (Data acquisition coupled to a specific hypothesis)*

*New model: "Download the world" (Data acquisition supports many hypotheses)*

- Astronomy: High-resolution, high-frequency sky surveys (SDSS, LSST, PanSTARRS)
- Biology: lab automation, high-throughput sequencing,
- Oceanography: high-resolution models, cheap sensors, satellites





# Example: Assessing Treatment Efficacy



	A	B	C	D	E	F	G	H	I	J
1	fu_2wk	fu_4wk	fu_8wk	fu_12wk	fu_16wk	fu_20wk	fu_24wk	total4type_fu	clinic_zip	pt_zip
2	1	3	4	7	9	9	9	12	98405	98405
3	2	4	6	7	8	8	8	8	98405	98403
4	0	0	0	0	0	0	0	0	98405	98445
5	3	2	2	2	2	5	5	5	98405	98332
6	0	0	0	0	0	0	0	8	98405	98405
7	2	2	2	2	2	2	2	2	98405	98402
8	1	2	5	6	8	10	10	14	98405	98418
9	1	1	2	2	2	2	2	2	98499	98406
10	0	0	0	0	0	0	0	0	98405	98404
11	0	0	0	0	0	0	0	0	98405	98402
12	1	1	1	1	1	1	1	1	98405	98405
13	1	1	1	1	1	1	1	1	98404	98404
14	2	2	2	2	2	2	2	2	98499	98498
15	0	0	0	0	0	0	0	0	98499	98445
16	1	2	4	5	7	7	7	7	98499	98405
17	1	1	1	2	2	2	2	2	98499	98498

number of follow ups within 16 weeks after treatment enrollment.

Zip code of clinic

Zip code of patient

*Question: Does the distance between the patient's home and clinic influence the number of follow ups, and therefore treatment efficacy?*

# Python Program to Assess Treatment Efficacy

```
# This program reads an Excel spreadsheet whose penultimate
# and antepenultimate columns are zip codes.
# It adds a new last column for the distance between those zip
# codes, and outputs in CSV (comma-separated values) format.
# Call the program with two numeric values: the first and last
# row to include.
# The output contains the column headers and those rows.
```

```
# Libraries to use
```

```
import random
import sys
import xlrd      # library for working with Excel spreadsheets
import time
from gdapi import GoogleDirections
```

```
# No key needed if few queries
```

```
gd = GoogleDirections('dummy-Google-key')
```

```
wb = xlrd.open_workbook('mhip_zip_eScience_121611a.xls')
sheet = wb.sheet_by_index(0)
```

```
# User input: first row to process, first row not to process
```

```
first_row = max(int(sys.argv[1]), 2)
row_limit = min(int(sys.argv[2])+1, sheet.nrows)
```

```
def comma_separated(lst):
    return ",".join([str(s) for s in lst])
```

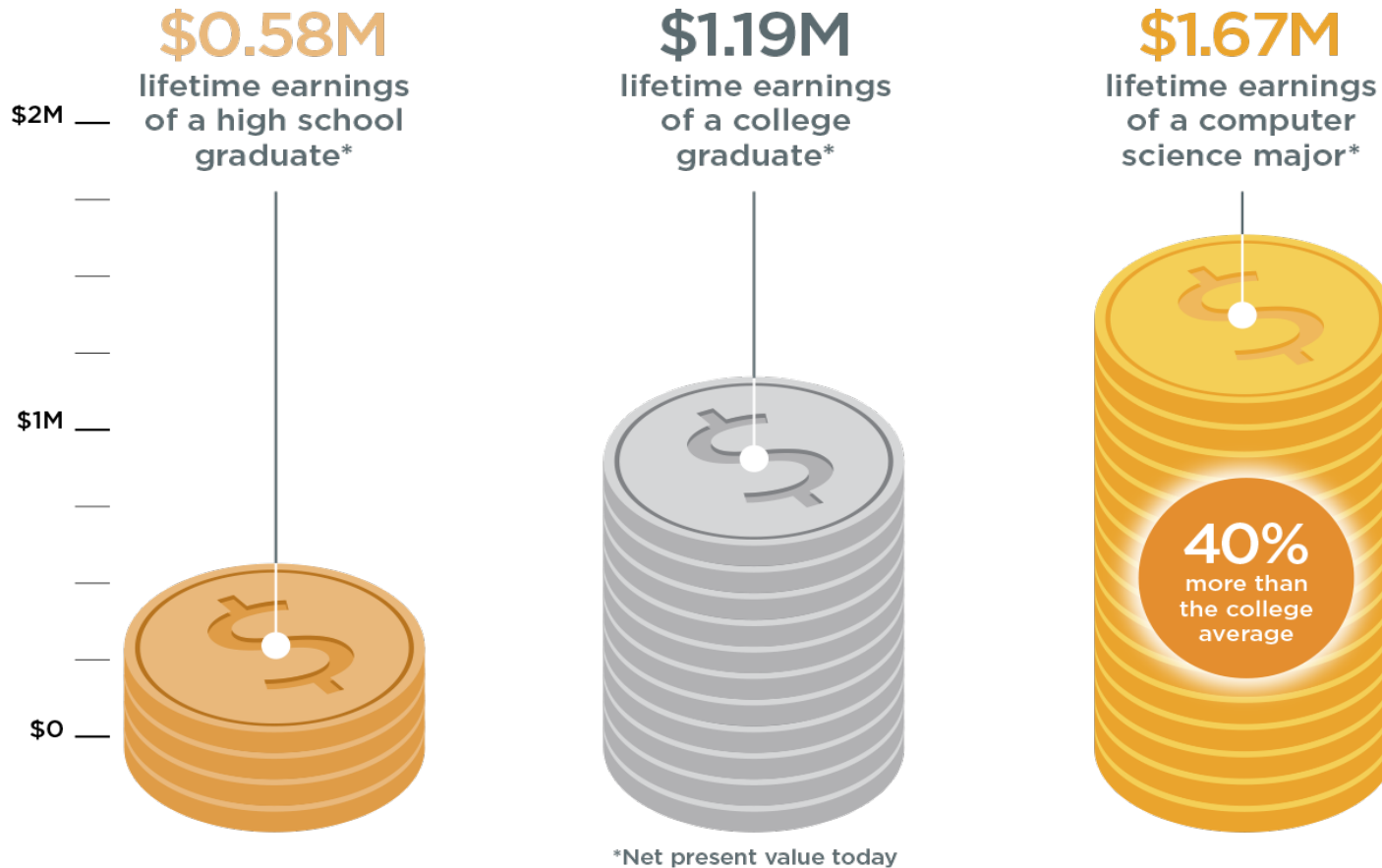
```
headers = sheet.row_values(0) + ["distance"]
print comma_separated(headers)

for rownum in range(first_row, row_limit):
    row = sheet.row_values(rownum)
    (zip1, zip2) = row[-3:-1]
    if zip1 and zip2:
        # Clean the data
        zip1 = str(int(zip1))
        zip2 = str(int(zip2))
        row[-3:-1] = [zip1, zip2]
        # Compute the distance via Google Maps
        try:
            distance = gd.query(zip1, zip2).distance
        except:
            print >> sys.stderr, "Error computing distance:", zip1,
            zip2
            distance = ""
        # Print the row with the distance
        print comma_separated(row + [distance])
        # Avoid too many Google queries in rapid succession
        time.sleep(random.random()+0.5)
```

23 lines of executable code!

# Some statistics (from U.S.)

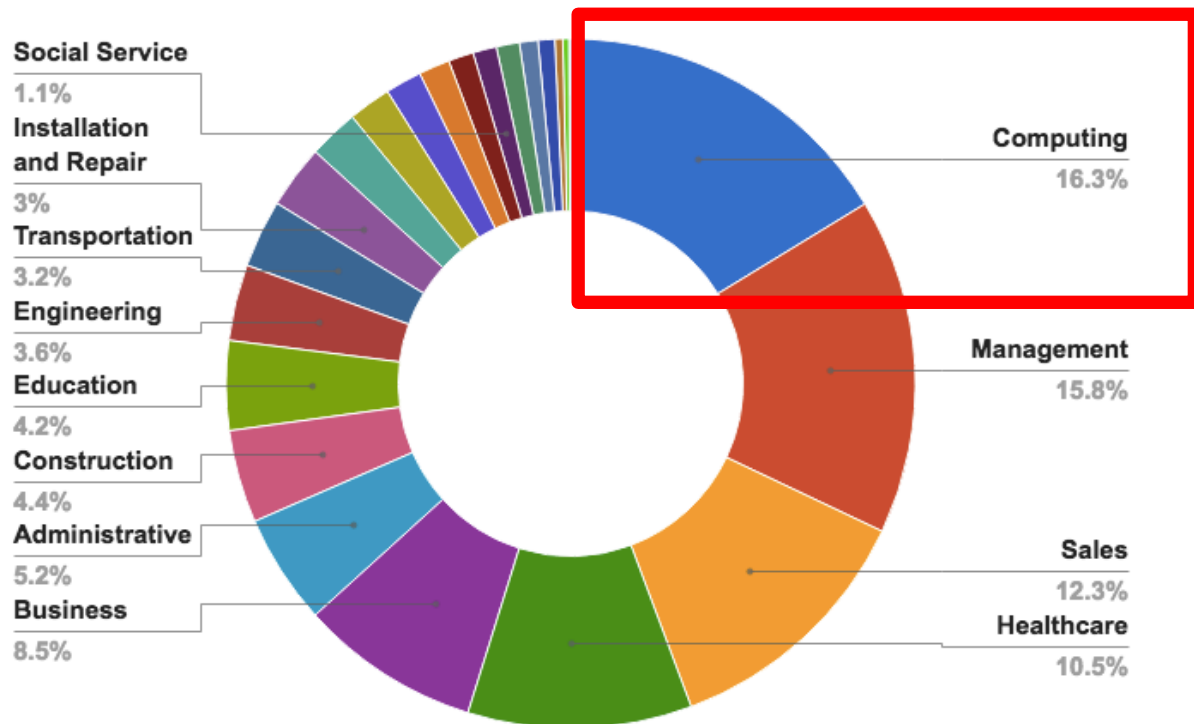
## The value of a computer science education



Source: Brookings

## Some statistics (from U.S.)

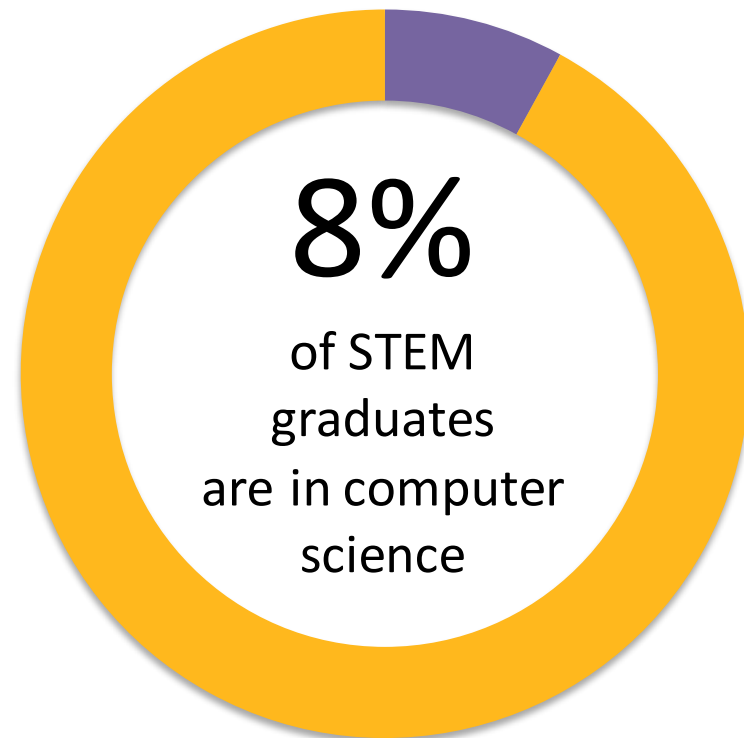
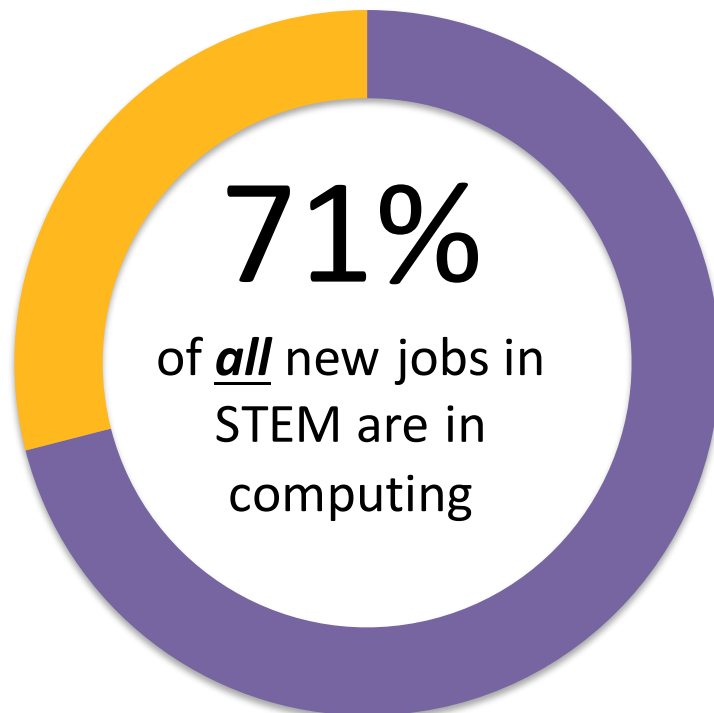
Computing jobs are the #1 source of new wages in the United States



500,000 current openings: These jobs are in **every** industry and **every** state, and they're projected to grow at twice the rate of all other jobs.

## Some statistics (from U.S.)

The STEM\* problem is in computer science:



Sources: Bureau of Labor Statistics, National Center for Education Statistics

# Course Logistics

- Website: <http://web.cs.hacettepe.edu.tr/~bbm101/>
- See the website for all administrative details
- Read the handouts and required texts, *before* the lecture
- Take notes!
- Follow the course in Piazza  
<https://piazza.com/hacettepe.edu.tr/fall12016/bbm101>

# Academic Integrity

- Honest work is required of a scientist or engineer.
- Collaboration policy on the course web. **Read it!**
  - Discussion is permitted.
  - **Carrying materials from discussion is not permitted.**
  - Everything you turn in must be your own work.
    - Cite your sources, explain any unconventional action.
  - **You may not view others' work.**
  - If you have a question, ask.
- We trust you completely.
- But we have no sympathy for trust violations – nor should you!

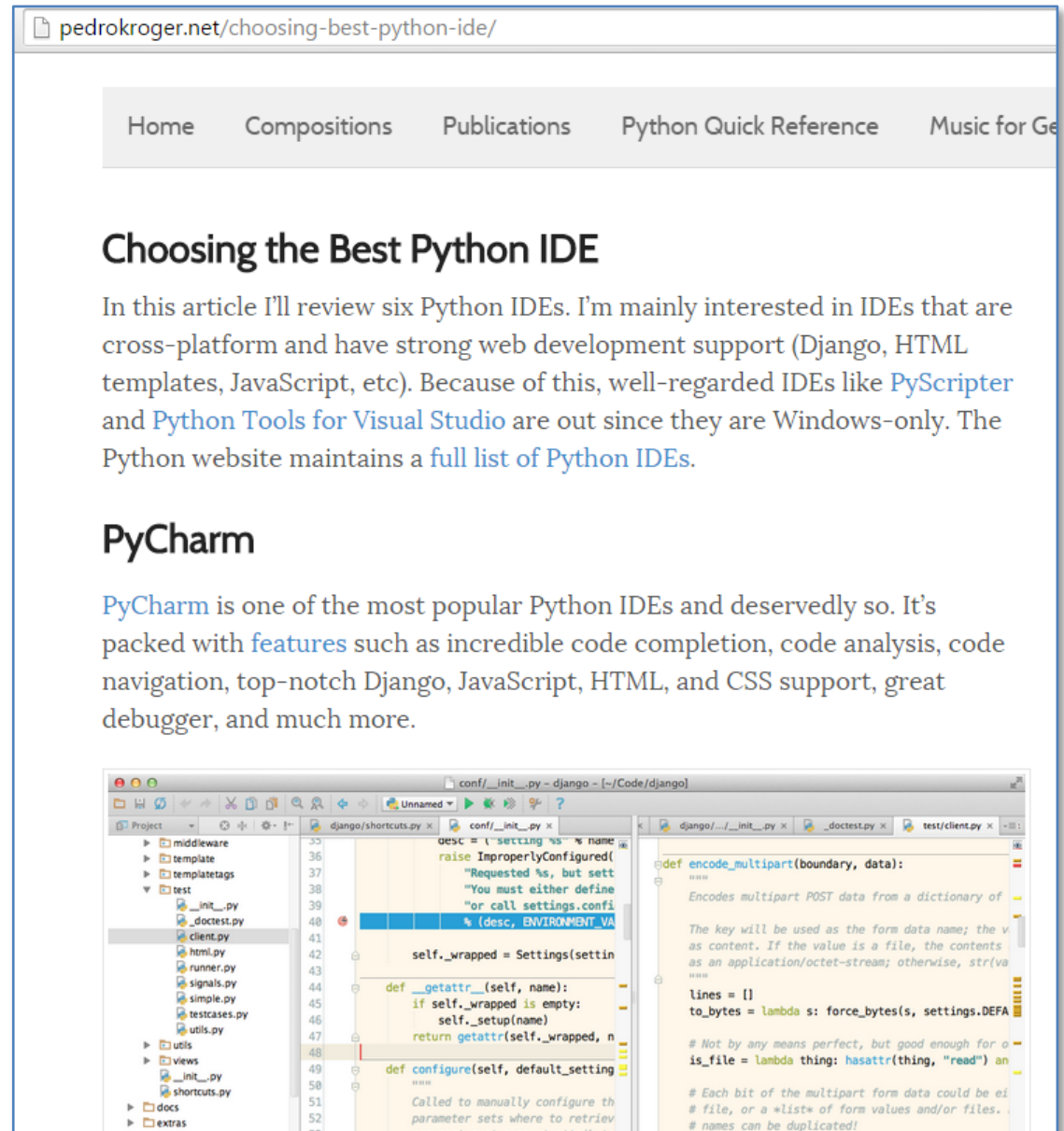


# How to Succeed

- No prerequisites
- Non-predictors for success:
  - Past programming experience
  - Enthusiasm for games or computers
- Programming and data analysis are challenging
- Every one of you can succeed
  - There is no such thing as a “**born programmer**”
  - Work hard
  - Follow directions
  - Be methodical
  - *Think* before you act
  - Try on your own, then ask for help
  - Start early

# Integrated Development Environment (IDE)

- There are many!



pedrokroger.net/choosing-best-python-ide/

Home Compositions Publications Python Quick Reference Music for Ge

## Choosing the Best Python IDE

In this article I'll review six Python IDEs. I'm mainly interested in IDEs that are cross-platform and have strong web development support (Django, HTML templates, JavaScript, etc). Because of this, well-regarded IDEs like [PyScripter](#) and [Python Tools for Visual Studio](#) are out since they are Windows-only. The Python website maintains a [full list of Python IDEs](#).

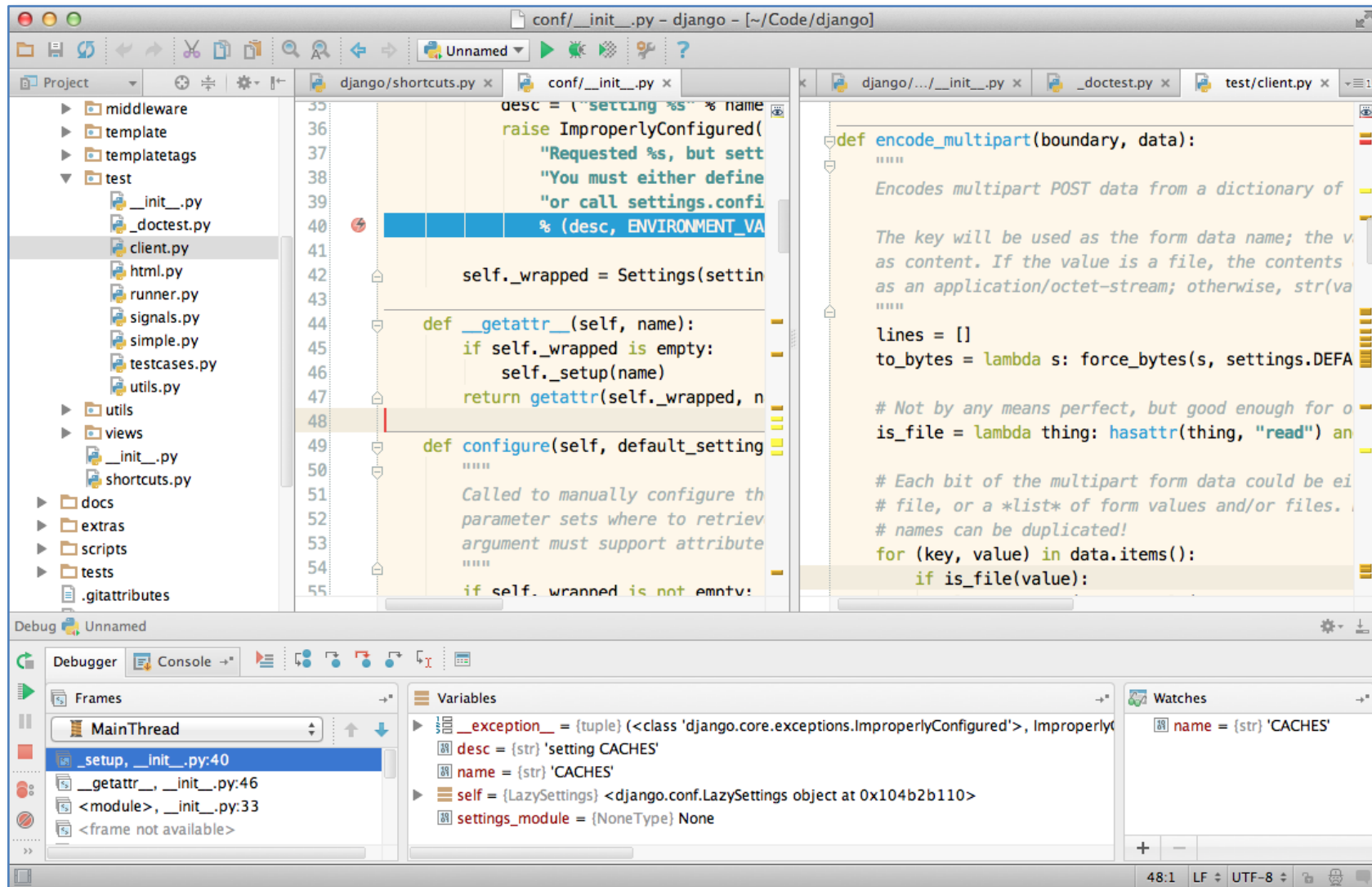
### PyCharm

[PyCharm](#) is one of the most popular Python IDEs and deservedly so. It's packed with [features](#) such as incredible code completion, code analysis, code navigation, top-notch Django, JavaScript, HTML, and CSS support, great debugger, and much more.

```
django/shortcuts.py x conf/_init_.py x
33 desc = """setting %s % name
36 raise ImproperlyConfigured(
37     "Requested %s, but sett
38     "You must either define
39     "or call settings.confi
40     % (desc, ENVIRONMENT_VA
41
42     self._wrapped = Settings(settin
43
44     def __getattr__(self, name):
45         if self._wrapped is empty:
46             self._setup(name)
47         return getattr(self._wrapped, n
48
49     def configure(self, default_setting
50
51     Called to manually configure th
52     parameter sets where to retriev
53     """
```

```
django/.../_init_.py x _doctest.py x test/client.py x
def encode_multipart(boundary, data):
    """
    Encodes multipart POST data from a dictionary of
    as content. If the value is a file, the contents
    as an application/octet-stream; otherwise, str(va
    """
    lines = []
    to_bytes = lambda s: force_bytes(s, settings.DEFA
    # Not by any means perfect, but good enough for o
    is_file = lambda thing: hasattr(thing, "read") an
    # Each bit of the multipart form data could be ei
    # file, or a »list» of form values and/or files.
    # names can be duplicated!
```

# Our Recommendation: PyCharm

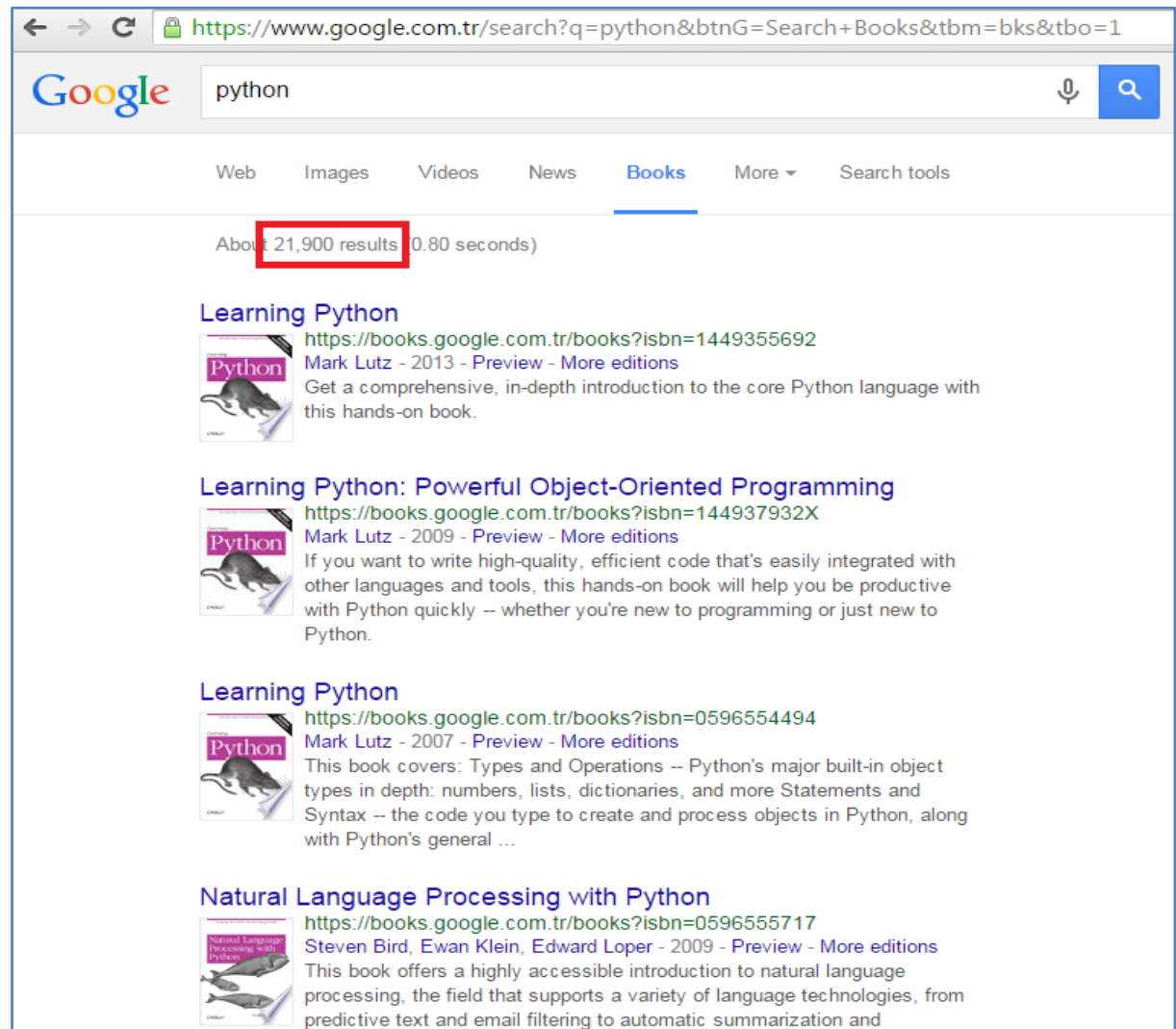


# Python Version

- Whatever IDE you choose to work with, always stick to **Python version 3.5.2**
- **Always** use this version to code your assignments.

# Books

- There are many!



The screenshot shows a Google search for 'python' on the Books tab. The search results are as follows:

- Learning Python**  
<https://books.google.com.tr/books?isbn=1449355692>  
Mark Lutz - 2013 - Preview - More editions  
Get a comprehensive, in-depth introduction to the core Python language with this hands-on book.
- Learning Python: Powerful Object-Oriented Programming**  
<https://books.google.com.tr/books?isbn=144937932X>  
Mark Lutz - 2009 - Preview - More editions  
If you want to write high-quality, efficient code that's easily integrated with other languages and tools, this hands-on book will help you be productive with Python quickly -- whether you're new to programming or just new to Python.
- Learning Python**  
<https://books.google.com.tr/books?isbn=0596554494>  
Mark Lutz - 2007 - Preview - More editions  
This book covers: Types and Operations -- Python's major built-in object types in depth: numbers, lists, dictionaries, and more Statements and Syntax -- the code you type to create and process objects in Python, along with Python's general ...
- Natural Language Processing with Python**  
<https://books.google.com.tr/books?isbn=0596555717>  
Steven Bird, Ewan Klein, Edward Loper - 2009 - Preview - More editions  
This book offers a highly accessible introduction to natural language processing, the field that supports a variety of language technologies, from predictive text and email filtering to automatic summarization and

# Our Recommendation for Books

- [The Python Tutorial](#), available from the Python website.
  - This is good for explaining the nuts and bolts of how Python works.
- [Introduction to Computation and Programming Using Python, Second Edition](#), John V. Guttag, MIT Press, August 2016
- [Think Python, 2nd edition](#)
  - Freely available online in [HTML](#) and [PDF](#).
  - Also available for purchase as a printed book, but don't buy the first edition.
  - This book introduces more conceptual material, motivating computational thinking.
- There is an [interactive version of “How to Think Like a Computer Scientist”](#) (the first edition of “Think Python”), which lets you type and run Python code directly while reading the book.