

# Programmazione 1

Introduzione al corso

Stefano Gualandi

Per ricerca: @famo2spaghi

Per didattica: @famo2conti

# Due questionari

1. Raccolta dati per creazione account
2. Questionario conoscitivo perchè capire le vostre conoscenze in ambito informatico
3. Andare all'indirizzo:  
<http://mate.unipv.it/gualandi/programming/>

# Informazioni generali

- **Siti del corso**

- <http://matematica.unipv.it/gualandi/programming>

- <https://github.com/mathcoding/programming>

- **Orario lezioni:**

- Mercoledì 12h00/13h00 – Aula C3

- Giovedì 11h00/13h00 – Aula B2

- Venerdì 11h00/13h00 – Aula C3 o Laboratorio didattico

- **Tutorato:** sarà attivato a partire da metà novembre

- **Esame:**

- Prova di programmazione in aula informatica

- Durata 2 ore

- **Ricevimento:**

- In qualsiasi momento potete richiedere ricevimento via email

- Appena possibile stabiliamo un giorno e orario in cui non avete lezione

# Comunicazioni con il docente

- Usare sempre l'email di ateneo  
[nome.cognome@universitadipavia.it](mailto:nome.cognome@universitadipavia.it)
- Cercare di essere precisi nel fare le domande
- Non avere timore di venire a ricevimento

# Testi di riferimento

Slides e script usati a lezione  
(in italiano), reperibili sul sito

**Materiale didattico**

Gli script Python usati a lezioni sono continuamente aggiornati su GitHub al repository seguente: [Programmazione 1](#)

Lucidi usati a lezione (per i notebooks python consultare il sito su github):

- Lucidi di introduzione al corso ...

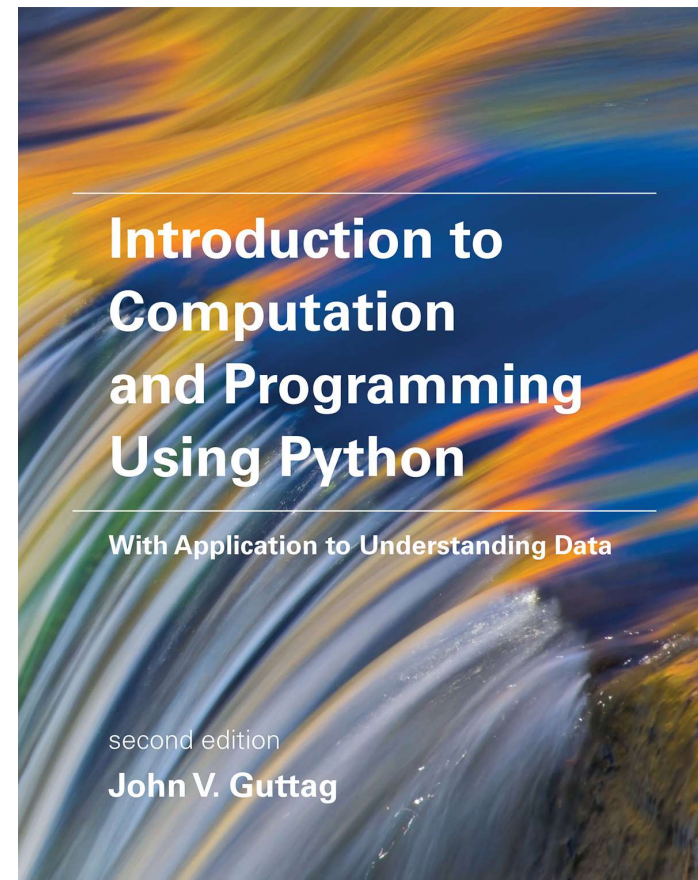
**Lecture Consigliate**

---

**Links Utili**

- Versione di Python consigliata: [Anaconda](#)
- Applicazione per connettersi a GitHub: [GitHub Desktop](#)
- [GitHub Student Pack](#)
- Documentazione ufficiale di Python 3.x: [docs](#)

Testo di riferimento (in inglese)



# Obiettivi del corso

1. Introdurre i concetti fondamentali di programmazione
2. Insegnare ad usare la programmazione come supporto alla risoluzione di problemi
3. Insegnare un linguaggio di programmazione multiparadigma (Python)
4. Stimolare la vostra curiosità e fantasia
5. Stimolare la vostra curiosità e fantasia
6. Stimolare la vostra curiosità e fantasia
- 7. *Coding is fun!***

# <http://movielens.org>

## movielens

Non-commercial, personalized movie recommendations.

[sign up now](#)

or [sign in](#)

### recommendations

MovieLens helps you find movies you will like. Rate movies to build a custom taste profile, then MovieLens recommends other movies for you to watch.

The screenshot displays the MovieLens interface with two main sections: 'top picks' and 'recent releases'. The 'top picks' section is titled 'based on your ratings, MovieLens recommends these movies' and features a row of movie cards. Each card includes the movie title, year, duration, and a star rating. The movies shown are 'Band of Brothers' (2001, 705 min, 8 stars), 'Casablanca' (1942, 102 min, 8 stars), 'One Flew Over the Cuckoo's Nest' (1975, 133 min, 8 stars), 'The Lives of Others' (2006, 137 min, 8 stars), 'Sunset Boulevard' (1950, 110 min, 8 stars), 'The Third Man' (1949, 104 min, 8 stars), and 'Pat' (1967). The 'recent releases' section is titled 'movies released in last 90 days that you haven't rated' and shows a row of movie cards including 'Carlin' (2014, 105 min, 8 stars), 'Felony' (2018, 8 stars), 'What If' (2014, 102 min, 8 stars), 'Frank' (2014, 96 min, 8 stars), 'Sin City: A Dame to Kill For' (2014, 102 min, 8 stars), 'If I Stay' (2014, 106 min, 8 stars), and 'Are' (2014).

**top picks** [see more](#)

based on your ratings, MovieLens recommends these movies

Band of Brothers	Casablanca	One Flew Over the Cuckoo's Nest	The Lives of Others	Sunset Boulevard	The Third Man	Pat
2001	1942	1975	2006	1950	1949	1967
705 min	102 min	133 min	137 min	110 min	104 min	
8	8	8	8	8	8	

**recent releases** [see more](#)

movies released in last 90 days that you haven't rated

Carlin'as	Felony	What If	Frank	Sin City: A Dame to Kill For	If I Stay	Are
2014	2018	2014	2014	2014	2014	2014
105 min		102 min	96 min	102 min	106 min	
8		8	8	8	8	

# Programma del corso

- Introduzione al linguaggio di programmazione Python
- Elementi di Programmazione funzionale
- Elementi di Programmazione procedurale
- Elementi di Programmazione orientata agli oggetti
- Cenni di complessità algoritmica
- Algoritmi di ricerca e di ordinamento
- Algoritmi di programmazione dinamica



# Risoluzione di problemi

- Formalismo per rappresentare un problema (e.g. grafi)
- Definizione formale del problema e degli obiettivi (e.g., quale funzione obiettivo?)
- Soluzione ammissibile, soluzione ottima
- Problema risolvibile/non risolvibile al calcolatore (complessità computazionale)
- Tempo di esecuzione vs. tempo di sviluppo



How the customer explained it



How the Project Leader understood it



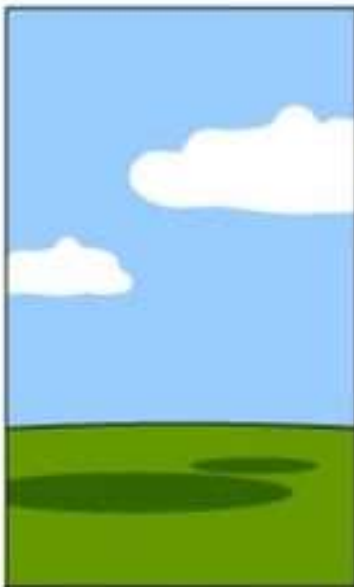
How the Analyst designed it



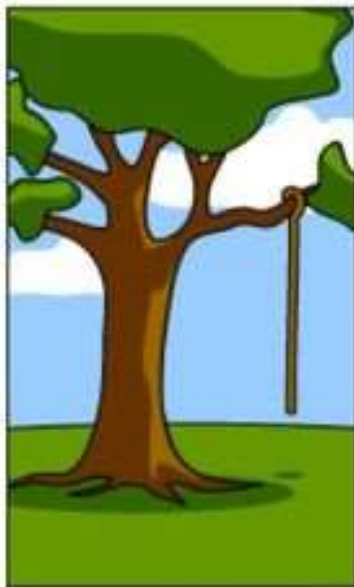
How the Programmer wrote it



How the Business Consultant described it



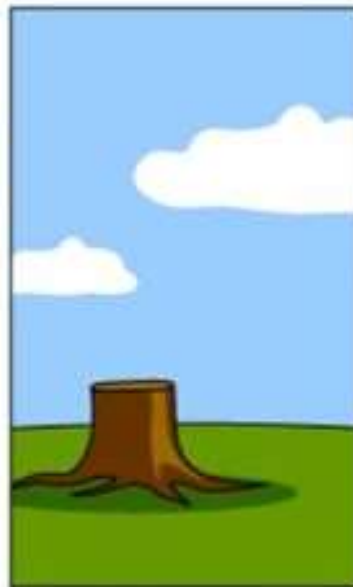
How the project was documented



What operations installed



How the customer was billed



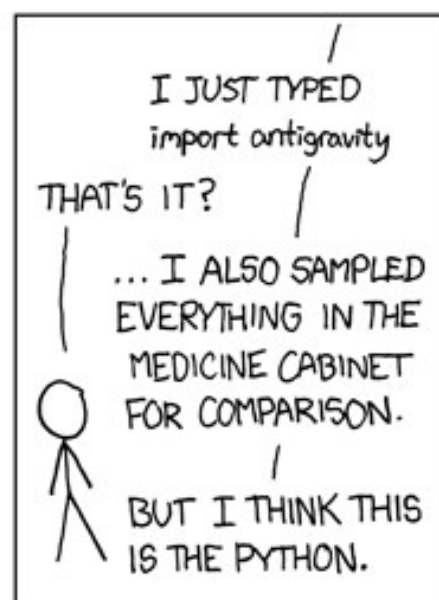
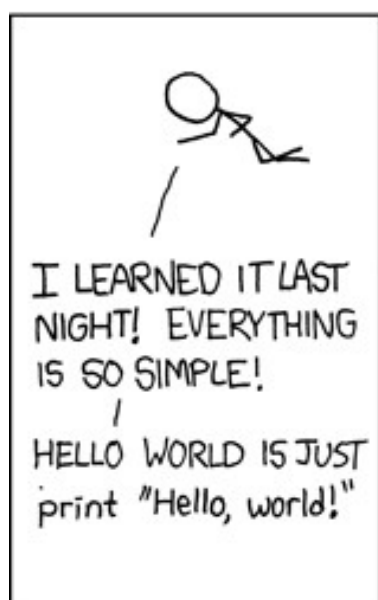
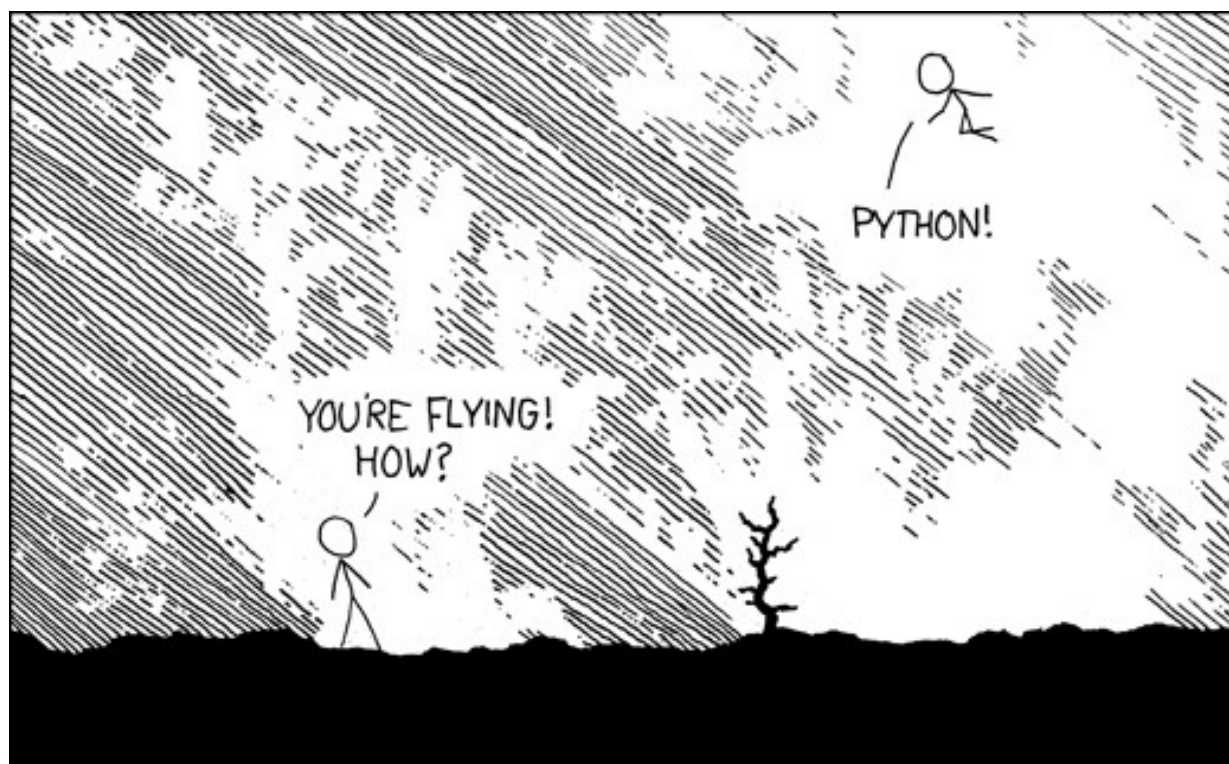
How it was supported

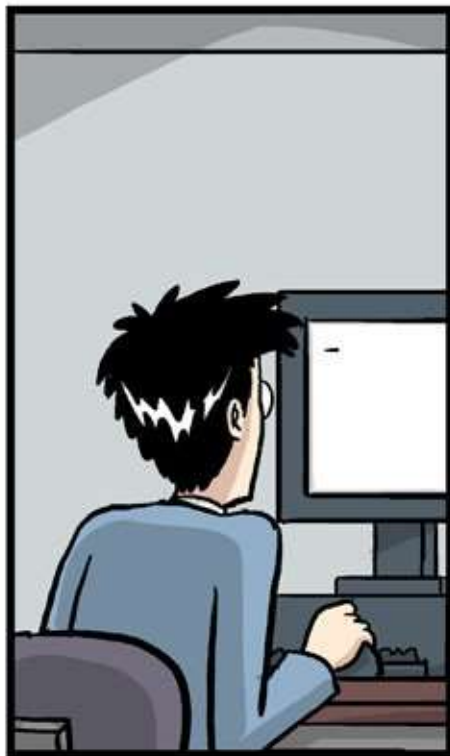


What the customer really needed

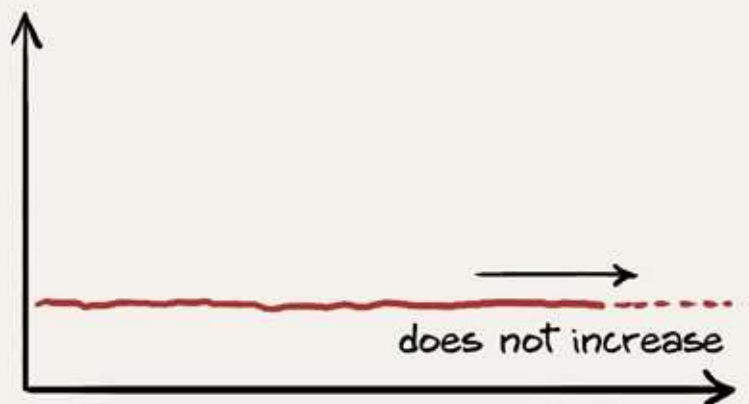
# Linguaggi di programmazione

- Perchè Python?
  - Semplice e divertente!
- Alternative:
  - C++ - <http://en.cppreference.com>
  - Julia - <http://julialang.org/>
  - Haskell - <http://www.haskell.org/>
  - R - <http://www.r-project.org/>
  - C# - **google-it-yourself**
  - Java - <http://www.java.com>





PROBABILITY YOU'LL COME UP  
WITH A BRILLIANT IDEA



TIME SPENT STARING AT YOUR  
COMPUTER

JORGE CHAM © 2012



ANY... MINUTE...  
NOW...

WWW.PHDCOMICS.COM

# Perchè Python?

## PRO:

- È semplice (no gestione diretta della memoria)
- È interattivo (interpretato, non compilato)
- È semplice *programmare* delle visualizzazioni
- È un linguaggio multiparadigma
- Ha un elevato numero di librerie semplici da installare

## CONTRO:

1. Non é efficiente come un linguaggio compilato
2. L'indentazione del codice fa parte della semantica
3. L'indentazione del codice fa parte della semantica
4. L'indentazione del codice fa parte della semantica

# Ambiente di lavoro Python

- A lezione e in laboratorio si farà riferimento a Python  $\geq 3.4$  (**attenzione a non scaricare o usare la versione di Python 2.7**)
- Ambiente di riferimento Anaconda/Python  $\geq 3.4$ :  
<https://www.continuum.io/downloads>
- Editors:
  - **CONSIGLIATO: Spyder3** (installato con Anaconda)
  - Vim o Emacs
  - Visual Studio (windows) o Visual Code
  - Sublime Text - <http://www.sublimetext.com/>
  - Rodeo - <http://www.yhat.com/products/rodeo>



*“E' l'istess...”*

*Spyder3 1/3*

The screenshot displays the Spyder Python IDE interface. The main editor window shows a Python script with the following content:

```
1# -*- coding: utf-8 -*-
2"""
3Created on Wed Oct 4 10:00:29 2017
4
5@author: gualandi
6"""
7
8
```

The Variable explorer panel on the right shows the following variables:

Name	Type	Size	Value
a	int	1	1
x	tuple	2	(1, 3)

The IPython console at the bottom shows the following output:

```
Python 3.5.3 |Anaconda custom (64-bit)| (default, Feb 22 2017, 21:28:42) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

Restarting kernel...

In [1]: a = 1
In [2]: x = (1, 3)
In [3]: print(x)
(1, 3)
In [4]:
```

The Windows taskbar at the bottom shows the system clock as 10:45 on 04/10/2017.



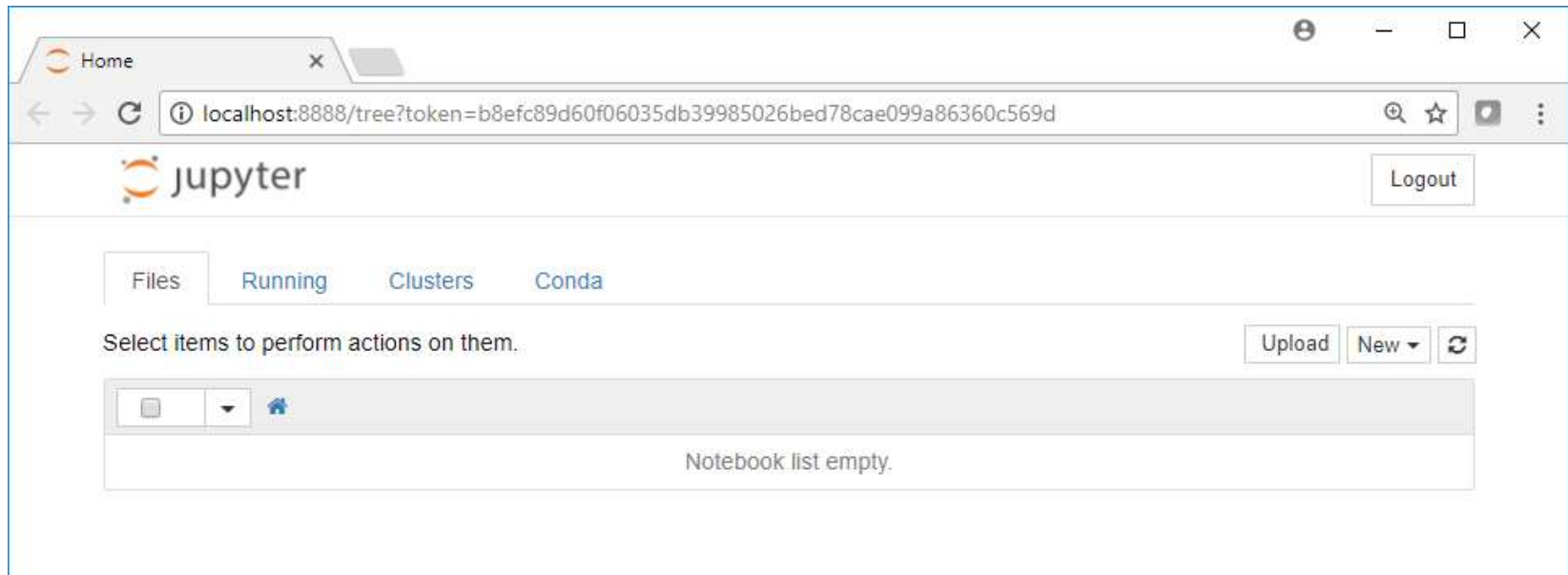
*“E' l'istess...”*

*Notebooks 2/3*

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

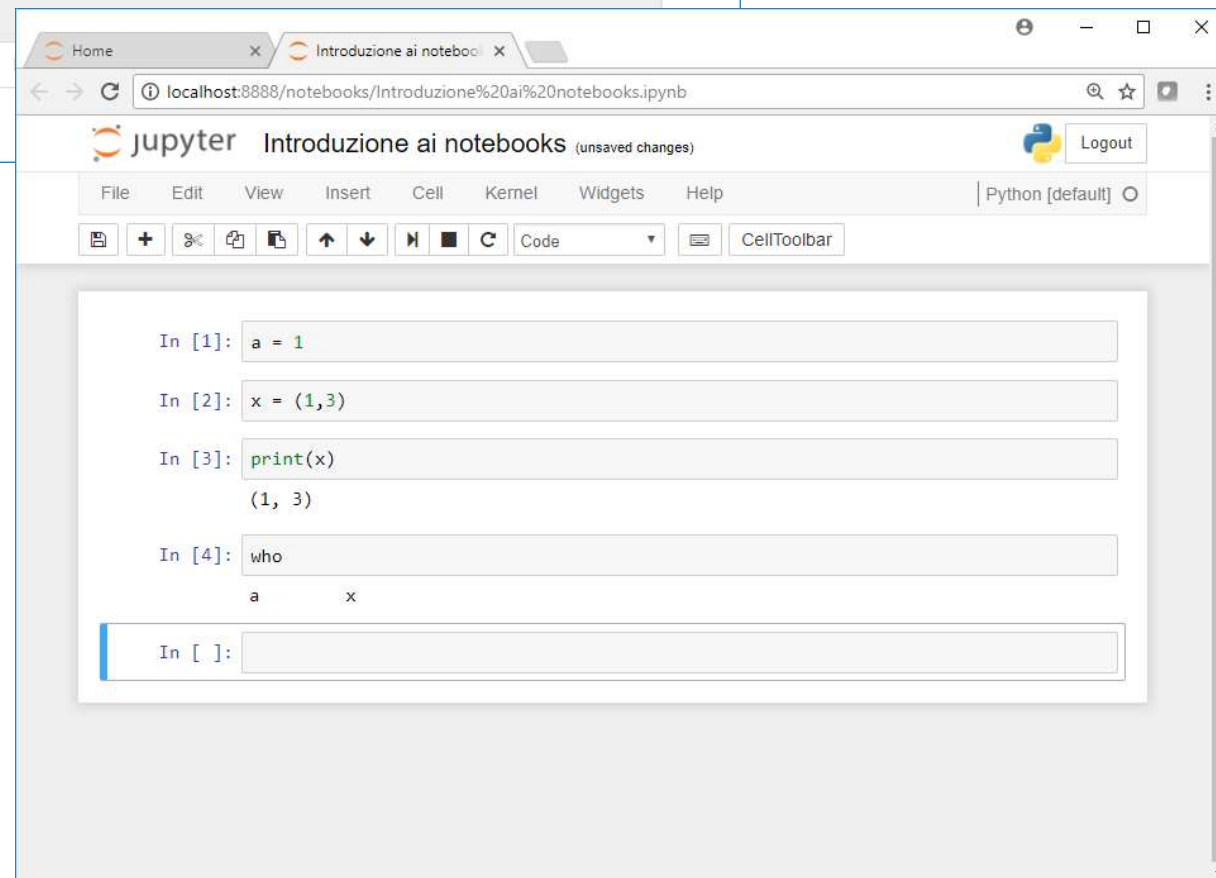
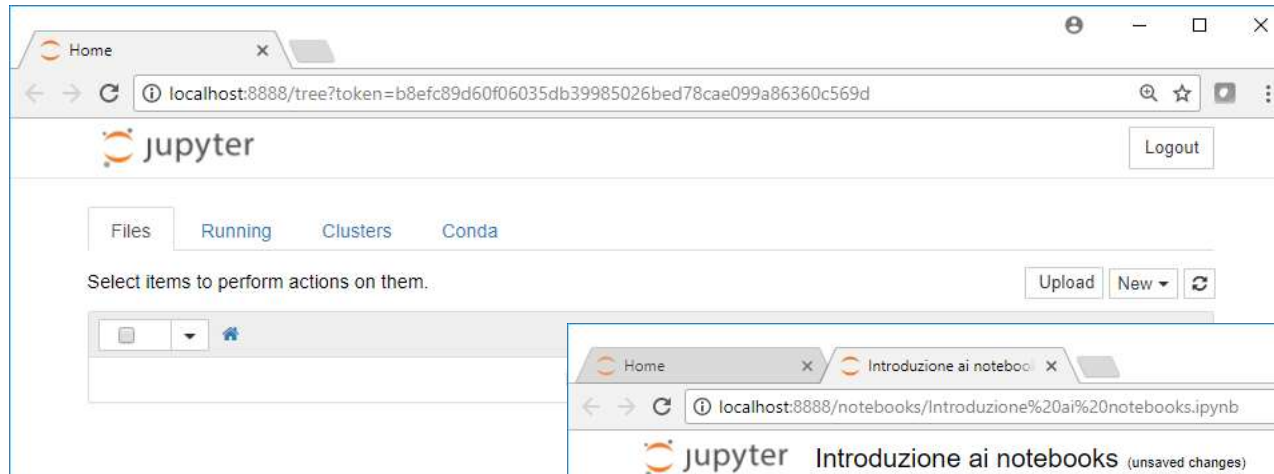
PS C:\Users\gualandi> cd D:\UnIPV\Lezioni\Programmazione1\notebooks\
PS D:\UnIPV\Lezioni\Programmazione1\notebooks> jupyter notebook
[I 10:49:37.857 NotebookApp] [nb_conda_kernels] enabled, 2 kernels found
[I 10:49:38.232 NotebookApp] [nb_anacondacloud] enabled
[I 10:49:38.232 NotebookApp] [nb_conda] enabled
[I 10:49:38.279 NotebookApp] \u2713 nbpresent HTML export ENABLED
[W 10:49:38.279 NotebookApp] \u2717 nbpresent PDF export DISABLED: No module named 'n
[I 10:49:38.357 NotebookApp] Serving notebooks from local directory: D:\UnIPV\Lezioni
[I 10:49:38.357 NotebookApp] 0 active kernels
[I 10:49:38.357 NotebookApp] The Jupyter Notebook is running at: http://localhost:888
[I 10:49:38.357 NotebookApp] Use Control-C to stop this server and shut down all kern
[C 10:49:38.373 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=84ce09cacc0a05df1c69830947ac5dc53c0c4fe05988d050
[I 10:49:38.525 NotebookApp] Accepting one-time-token-authenticated connection from :
```



*“E' l'istess...”*

2/3



*“E' l'istess...”*

3/3

```
stegua@DESKTOP-MH2Q2VS: ~  
stegua@DESKTOP-MH2Q2VS:~$ python3.5  
Python 3.5.2 (default, Nov 17 2016, 17:05:23)  
[GCC 5.4.0 20160609] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> a = 1  
>>> x = (1, 3)  
>>> print(x)  
(1, 3)  
>>> exit()  
stegua@DESKTOP-MH2Q2VS:~$
```

```
Windows PowerShell  
Copyright (C) 2016 Microsoft Corporation. All rights reserved.  
  
PS C:\Users\gualandi> ipython  
Python 3.5.3 |Anaconda custom (64-bit)| (default, Feb 22 2017, 21:28:42) [MSC  
Type "copyright", "credits" or "license" for more information.  
  
IPython 5.1.0 -- An enhanced Interactive Python.  
?                -> Introduction and overview of IPython's features.  
%quickref         -> Quick reference.  
help              -> Python's own help system.  
object?           -> Details about 'object', use 'object??' for extra details.  
  
In [1]: a = 1  
  
In [2]: x = (1,3)  
  
In [3]: print(x)  
(1, 3)  
  
In [4]: who  
a        x  
  
In [5]: exit()  
PS C:\Users\gualandi>
```

# Corsi online (self-paced)

- Coursera:

<http://www.coursera.org/learn/python>

- Udacity:

<http://www.udacity.com/course/programming-foundations-with-python--ud036>

- EDX:

<http://www.edx.org/course/learn-program-using-python-utarlingtonx-cse1309x>

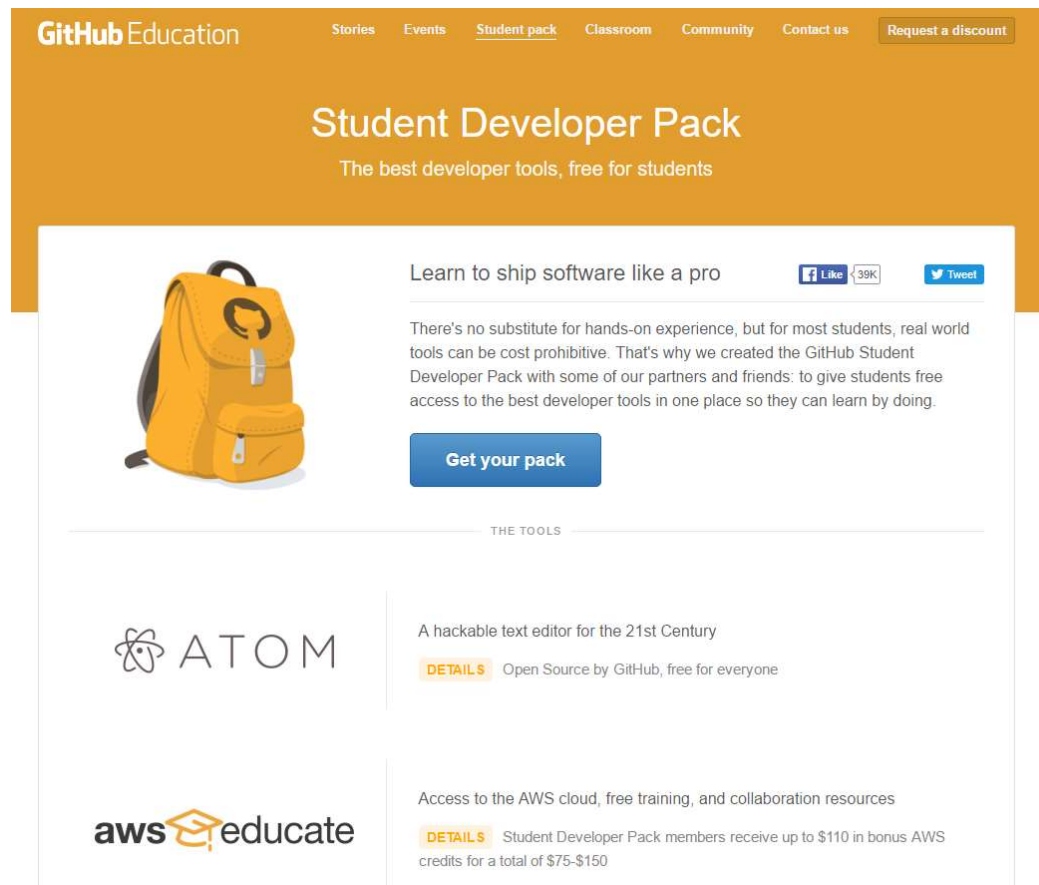
- Learn to code the hard way:

<http://learncodethehardway.org/python/>

# Altre risorse utili 1/2

## GitHub Education Pack:

<http://education.github.com/pack>




The screenshot shows the GitHub Education website's 'Student Developer Pack' page. The header is orange with the 'GitHub Education' logo and navigation links: 'Stories', 'Events', 'Student pack' (underlined), 'Classroom', 'Community', 'Contact us', and a 'Request a discount' button. The main heading is 'Student Developer Pack' with the tagline 'The best developer tools, free for students'. Below this is a white box containing an illustration of a yellow backpack with a GitHub logo. To the right of the backpack, the text reads 'Learn to ship software like a pro' with social media share buttons for Facebook (39K likes) and Twitter. A paragraph explains the pack's purpose: 'There's no substitute for hands-on experience, but for most students, real world tools can be cost prohibitive. That's why we created the GitHub Student Developer Pack with some of our partners and friends: to give students free access to the best developer tools in one place so they can learn by doing.' A blue 'Get your pack' button is positioned below the text. Under the heading 'THE TOOLS', two featured tools are listed: 'ATOM' (a hackable text editor) and 'aws educate' (AWS cloud access and training). Each tool entry includes a 'DETAILS' link and a brief description of the benefits for pack members.

GitHub Education

Stories Events Student pack Classroom Community Contact us Request a discount

## Student Developer Pack

The best developer tools, free for students



Learn to ship software like a pro [Like](#) [39K](#) [Tweet](#)

There's no substitute for hands-on experience, but for most students, real world tools can be cost prohibitive. That's why we created the GitHub Student Developer Pack with some of our partners and friends: to give students free access to the best developer tools in one place so they can learn by doing.

[Get your pack](#)


### THE TOOLS



ATOM

A hackable text editor for the 21st Century

[DETAILS](#) Open Source by GitHub, free for everyone



aws educate

Access to the AWS cloud, free training, and collaboration resources

[DETAILS](#) Student Developer Pack members receive up to \$110 in bonus AWS credits for a total of \$75-\$150

# Altre risorse utili 2/2

## Amazon Web Services educate:

<http://aws.amazon.com/education/awseducate/>



### Institutions

Provide educators and students with resources for cloud-related learning. Those at member institutions receive twice as many AWS credits, demos and special on-campus programs.

[Apply for AWS Educate for Institutions »](#)

[Already a Member?](#)



### Educators

Professors, teaching assistants, and educators receive access to AWS technology, open source content for their courses, training resources, and a community of cloud evangelists.

[Apply for AWS Educate for Educators »](#)

[Already a Member?](#)



### Students

Students receive credits for hands-on experience with AWS technology, training, content, career pathways, and job board.

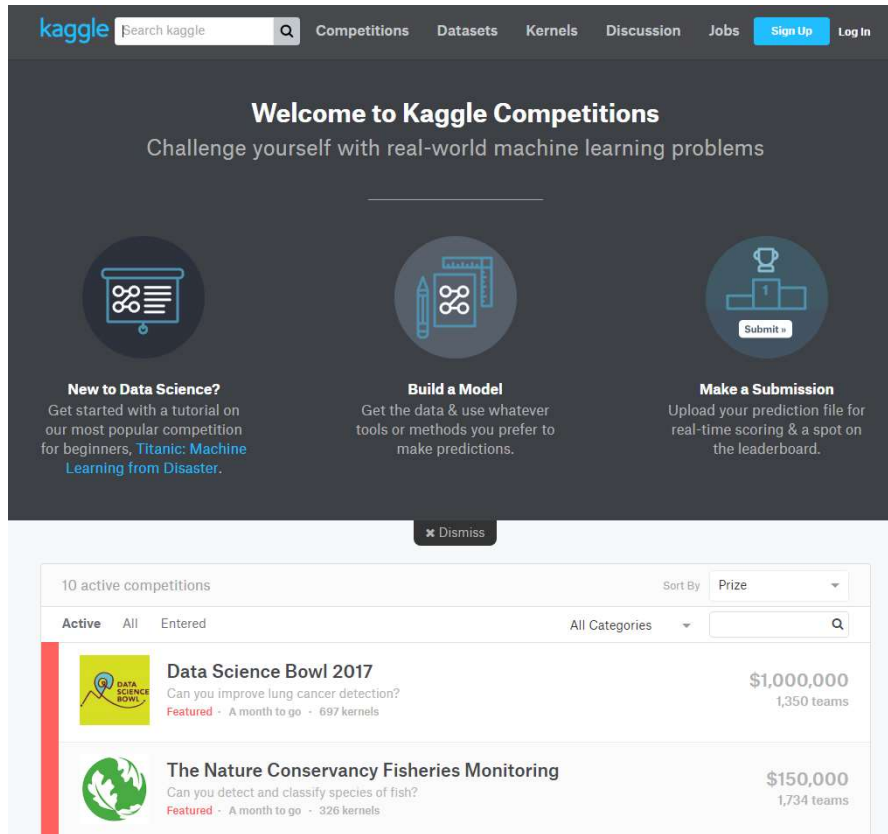
[Apply for AWS Educate for Students »](#)

[Already a Member? Check your welcome email.](#)



# Competizioni

- <http://www.kaggle.com>
- <http://www.crowdanalytix.com>



The image shows the Kaggle Competitions homepage. At the top, there is a navigation bar with the Kaggle logo, a search bar, and links to Competitions, Datasets, Kernels, Discussion, Jobs, Sign Up, and Log In. The main heading is "Welcome to Kaggle Competitions" with the subtitle "Challenge yourself with real-world machine learning problems". Below this, there are three circular icons representing different stages: "New to Data Science?", "Build a Model", and "Make a Submission". Each icon has a brief description and a link to a tutorial or guide. At the bottom, there is a section for "10 active competitions" with a list of competitions, including "Data Science Bowl 2017" and "The Nature Conservancy Fisheries Monitoring".

**Welcome to Kaggle Competitions**  
Challenge yourself with real-world machine learning problems

**New to Data Science?**  
Get started with a tutorial on our most popular competition for beginners, [Titanic: Machine Learning from Disaster](#).

**Build a Model**  
Get the data & use whatever tools or methods you prefer to make predictions.

**Make a Submission**  
Upload your prediction file for real-time scoring & a spot on the leaderboard.

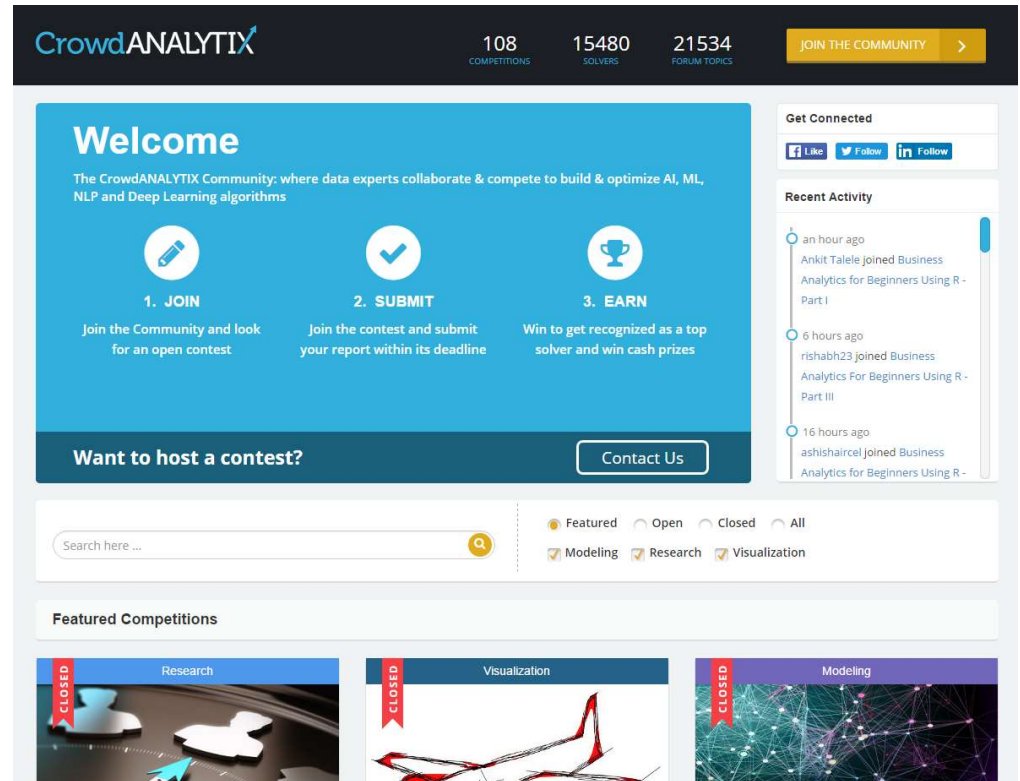
10 active competitions

Sort By: Prize

Active All Entered All Categories

**Data Science Bowl 2017**  
Can you improve lung cancer detection?  
Featured · A month to go · 697 kernels  
\$1,000,000  
1,350 teams

**The Nature Conservancy Fisheries Monitoring**  
Can you detect and classify species of fish?  
Featured · A month to go · 326 kernels  
\$150,000  
1,734 teams



The image shows the CrowdANALYTIX homepage. At the top, there is a navigation bar with the CrowdANALYTIX logo, statistics for 108 Competitions, 15480 Solvers, and 21534 Forum Topics, and a "JOIN THE COMMUNITY" button. The main heading is "Welcome" with the subtitle "The CrowdANALYTIX Community: where data experts collaborate & compete to build & optimize AI, ML, NLP and Deep Learning algorithms". Below this, there are three circular icons representing the steps: "1. JOIN", "2. SUBMIT", and "3. EARN". Each icon has a brief description and a link to a guide. At the bottom, there is a section for "Featured Competitions" with a list of competitions, including "Research", "Visualization", and "Modeling".

**Welcome**  
The CrowdANALYTIX Community: where data experts collaborate & compete to build & optimize AI, ML, NLP and Deep Learning algorithms

**1. JOIN**  
Join the Community and look for an open contest

**2. SUBMIT**  
Join the contest and submit your report within its deadline

**3. EARN**  
Win to get recognized as a top solver and win cash prizes

Want to host a contest? [Contact Us](#)

Get Connected  
[Like](#) [Follow](#) [Follow](#)

Recent Activity

- an hour ago  
Ankit Talele joined Business Analytics for Beginners Using R - Part I
- 6 hours ago  
rishabh23 joined Business Analytics For Beginners Using R - Part III
- 16 hours ago  
ashishaircel joined Business Analytics for Beginners Using R -

Search here ...

Featured Open Closed All  
Modeling Research Visualization


**Featured Competitions**

**Research**  
CLOSED

**Visualization**  
CLOSED

**Modeling**  
CLOSED

# SymPy: Symbolic Mathematics

 SymPy

[Main Page](#) [Features](#) [Download](#) [Documentation](#) [Support](#) [Development](#) [Donate](#) [Online Shell](#)

## About

SymPy is a Python library for symbolic mathematics. It aims to become a full-featured computer algebra system (CAS) while keeping the code as simple as possible in order to be comprehensible and easily extensible. SymPy is written entirely in Python.

[Get started with the tutorial](#) [Download Now](#)

## Why SymPy

SymPy is...

- **Free:** Licensed under BSD, SymPy is free both as in speech and as in beer.
- **Python-based:** SymPy is written entirely in Python and uses Python for its language.
- **Lightweight:** SymPy only depends on [mpmath](#), a pure Python library for arbitrary floating point arithmetic, making it easy to use.
- **A library:** Beyond use as an interactive tool, SymPy can be embedded in other applications and extended with custom functions.

[See SymPy's features](#)

## Projects using SymPy

This is an (incomplete) list of projects that use SymPy. If you use SymPy in your project, please let us know on our [mailinglist](#), so that we can add your project here as well.

- **Cadabra:** Tensor algebra and (quantum) field theory system using SymPy for scalar algebra.
- **SageMath:** Open source mathematics system that includes SymPy.
- **PyDy:** Multibody dynamics in Python.
- **galgebra:** Geometric algebra (previously [sympy.galgebra](#)).
- **yt:** Python package for analyzing and visualizing volumetric data ([yt.units](#) uses SymPy).

## Compute with Gamma

[Compute](#)



## Download Now

[Latest Version](#)  
[Development Version](#)

## Quick Links

- [Documentation](#)
- [Downloads \(source tarballs\)](#)
- [Mailing list](#)
- [Source code](#)
- [Issues tracker](#)
- [Wiki](#)
- [Introduction to contributing](#)
- [Try SymPy online now](#)
- [Official SymPy blog](#)
- [Planet SymPy](#)
- [Chat \(Gitter\)](#)
- [IRC channel logs](#)

## Follow Us

 [Follow](#) 527  [Follow @sympy](#)