

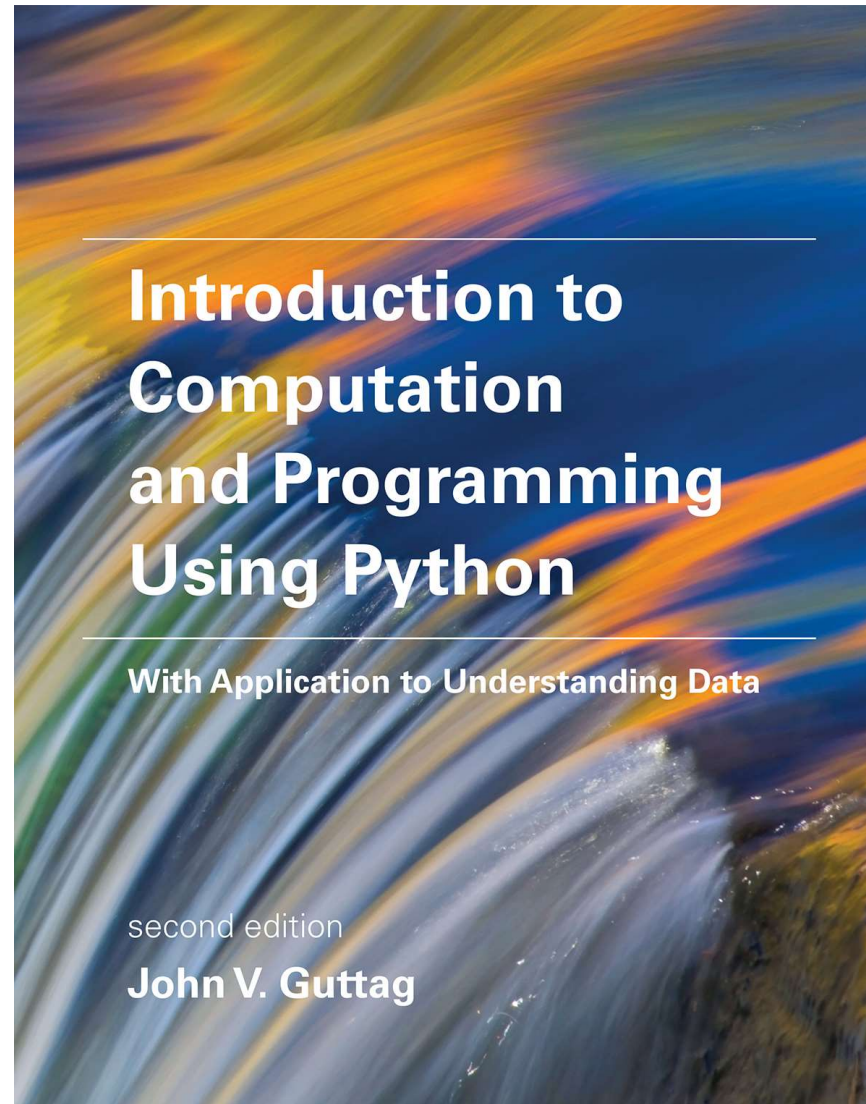
Programmazione 2

Introduzione al corso

Informazioni generali

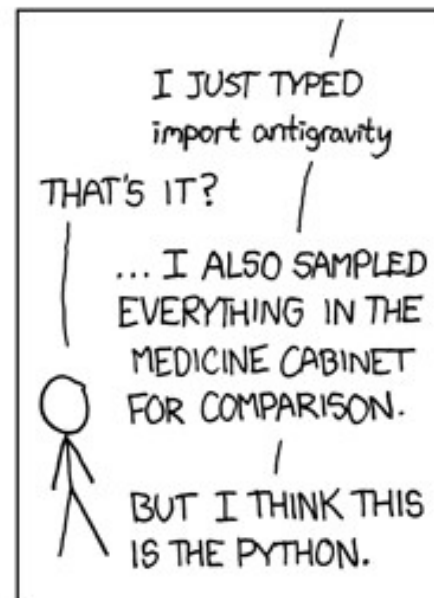
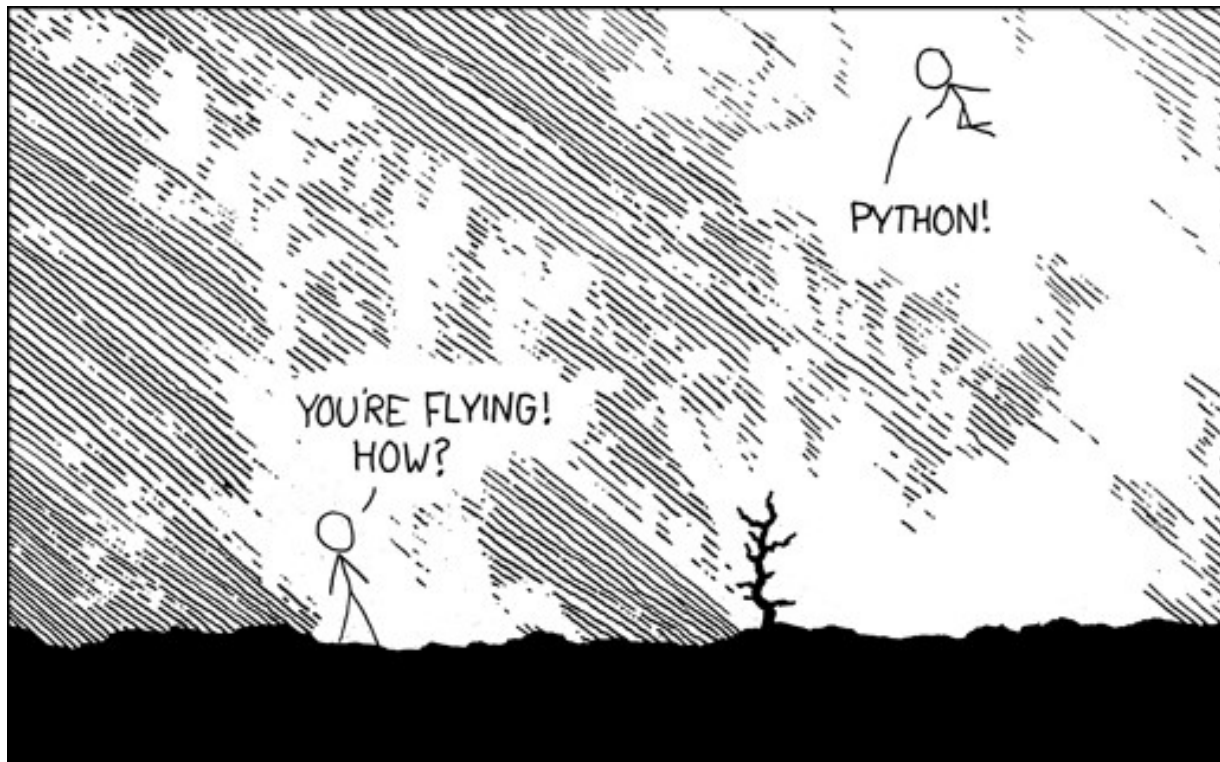
- Sito del corso
<http://matematica.unipv.it/gualandi/programmazione2/>
- Orario lezioni:
 - Martedì 9h00/11h00 – Aula C8 o Lab. Mate
 - Giovedì 14h00/15h00 – Aula C8 o Lab. Mate (non sempre)
- Breve ciclo di seminari tenuti da aziende
- Esami:
 - Idoneità (... no voto!)
 - Laboratori valutati
 - Modalità prova d'esame ANCORA da stabilire
- Ricevimento:
 - Quando volete, ma richiedere via email (con domande precise!)

Testo di riferimento



Obiettivi del corso

1. Offrire una panoramica su diversi 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!



Titanic



<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 grid of movie cards. Each card includes the movie title, year, and a star rating. The 'recent releases' section is titled 'movies released in last 90 days that you haven't rated' and also features a grid of movie cards with titles, years, and star ratings.

Movie Title	Year	Star Rating
Band of Brothers	2001	4.5
Casablanca	1942	4.5
One Flew Over the Cuckoo's Nest	1975	4.5
The Lives of Others	2006	4.5
Sunset Boulevard	1950	4.5
The Third Man	1949	4.5
Pat	1957	4.5
Carlin/Ras	2014	4.5
Felony	2016	4.5
What If	2014	4.5
Frank	2014	4.5
Sin City: A Dame to Kill For	2014	4.5
If I Stay	2014	4.5
Are You a Dummy?	2014	4.5

Paradigmi di programmazione

- Conoscenza dichiarativa vs. conoscenza imperativa
- Programmazione funzionale
- Programmazione ad oggetti
- Programmazione concorrente

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



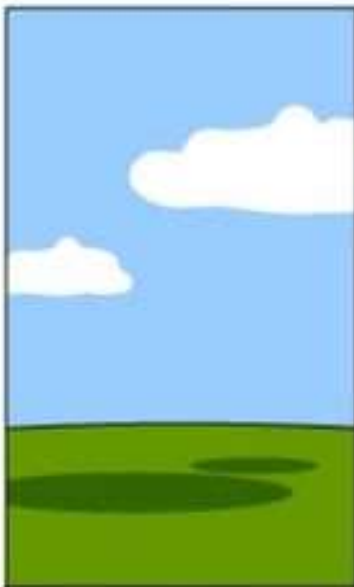
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



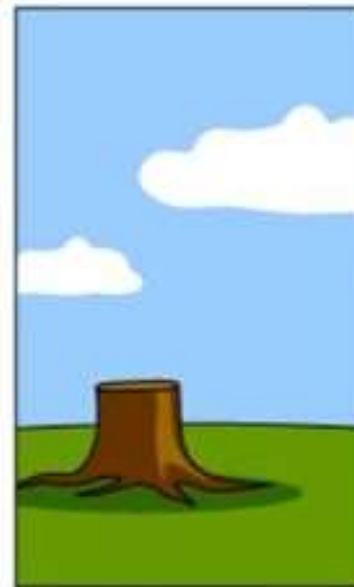
How the project was documented



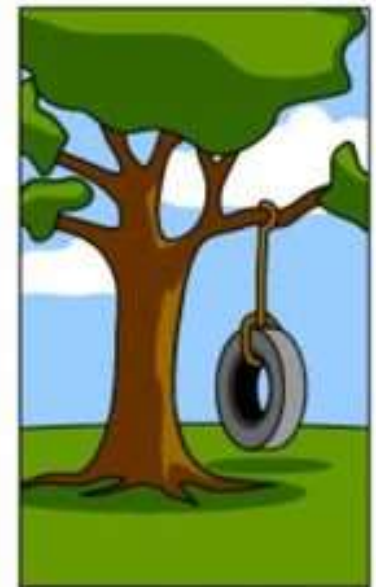
What operations installed



How the customer was billed



How it was supported



What the customer really needed

Classificazione linguaggi programmazione

- Basso livello vs. Alto livello
 - Livello di astrazione a cui si riferiscono le **primitive** del linguaggio
- General vs. Application domain
 - Linguaggio pensato per uno specific dominio applicativo (e.g., SQL o VHDL) o per un uso “general purpose”
- Interpretato vs. Compilato
 - Se il codice sorgente viene eseguito direttamente o viene prima compilato il linguaggio macchina

Linguaggi di programmazione

- Perché 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>

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

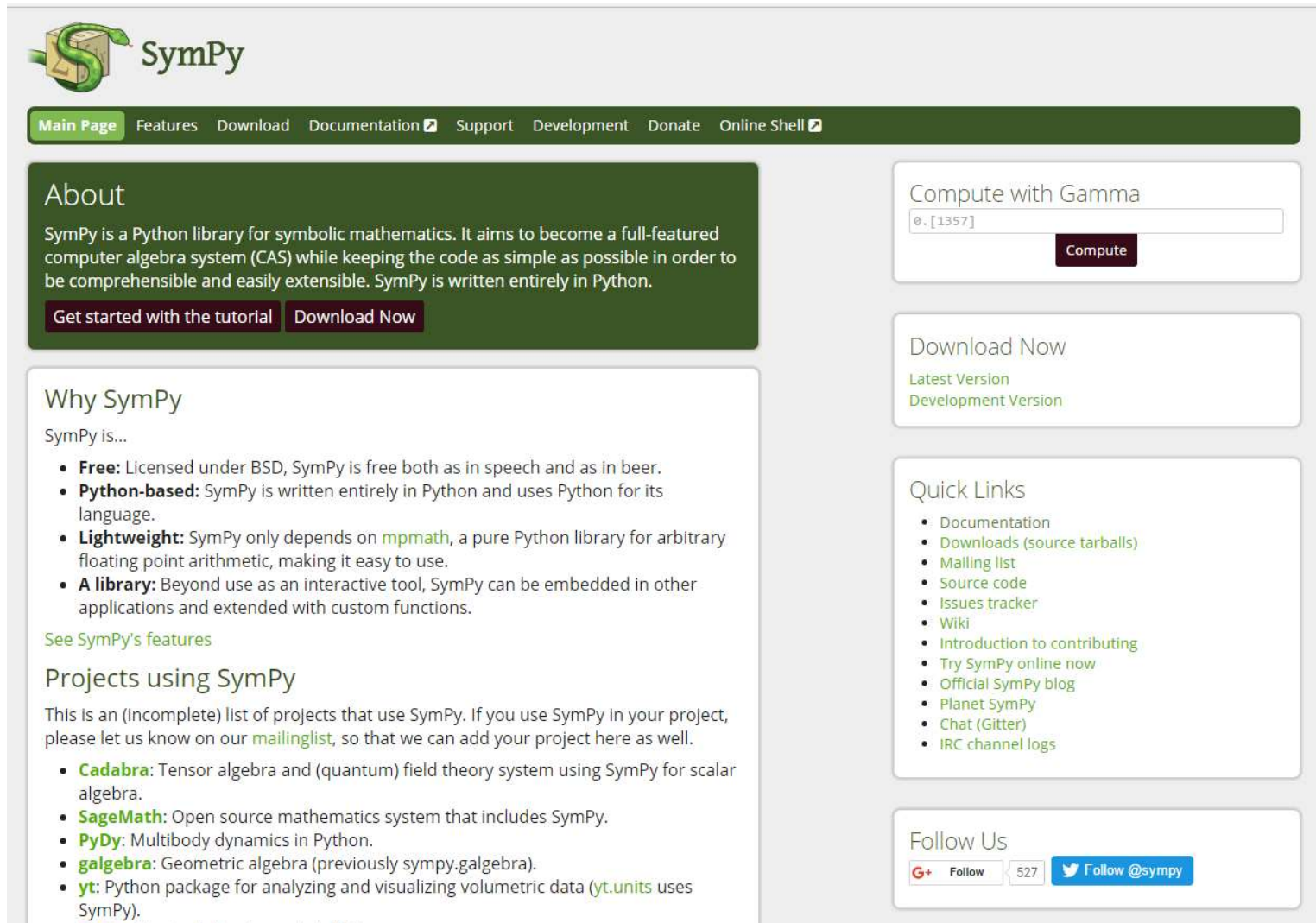
CONTRO:

- Non é efficiente come un linguaggio compilato
- L'indentazione del codice fa parte della semantica
- L'indentazione del codice fa parte della semantica
- L'indentazione del codice fa parte della semantica


Ambiente di lavoro Python

- A lezione e in laboratorio si farà riferimento a Python ≥ 3.5 (attenzione a non scaricare la versione 2.7)
- Ambiente di riferimento (Python 3.6):
<https://www.continuum.io/downloads>
- Utilizzo di notebooks (vedi demo)
- Editors:
 - Spyder (installato in automatico con Anaconda)
 - Vim o Emacs
 - Visual Studio (windows) o Visual Code
 - Sublime Text - <http://www.sublimetext.com/>
 - Rodeo - <http://www.yhat.com/products/rodeo>

SymPy: Symbolic Mathematics



The screenshot shows the SymPy website homepage. At the top left is the SymPy logo, which consists of a green cube with a white 'S' and a green snake-like shape. To the right of the logo is the text 'SymPy'. Below the logo is a dark green navigation bar with white text for 'Main Page', 'Features', 'Download', 'Documentation', 'Support', 'Development', 'Donate', and 'Online Shell'. The main content area is divided into several sections. On the left, there is a dark green 'About' section with white text describing SymPy as a Python library for symbolic mathematics. Below this is a 'Why SymPy' section with a list of bullet points. On the right, there is a 'Compute with Gamma' section with a text input field containing '0. [1357]' and a 'Compute' button. Below that is a 'Download Now' section with links for 'Latest Version' and 'Development Version'. Further down is a 'Quick Links' section with a list of links. At the bottom right is a 'Follow Us' section with buttons for 'Follow' (with a count of 527) and 'Follow @sympy'.

 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](#)

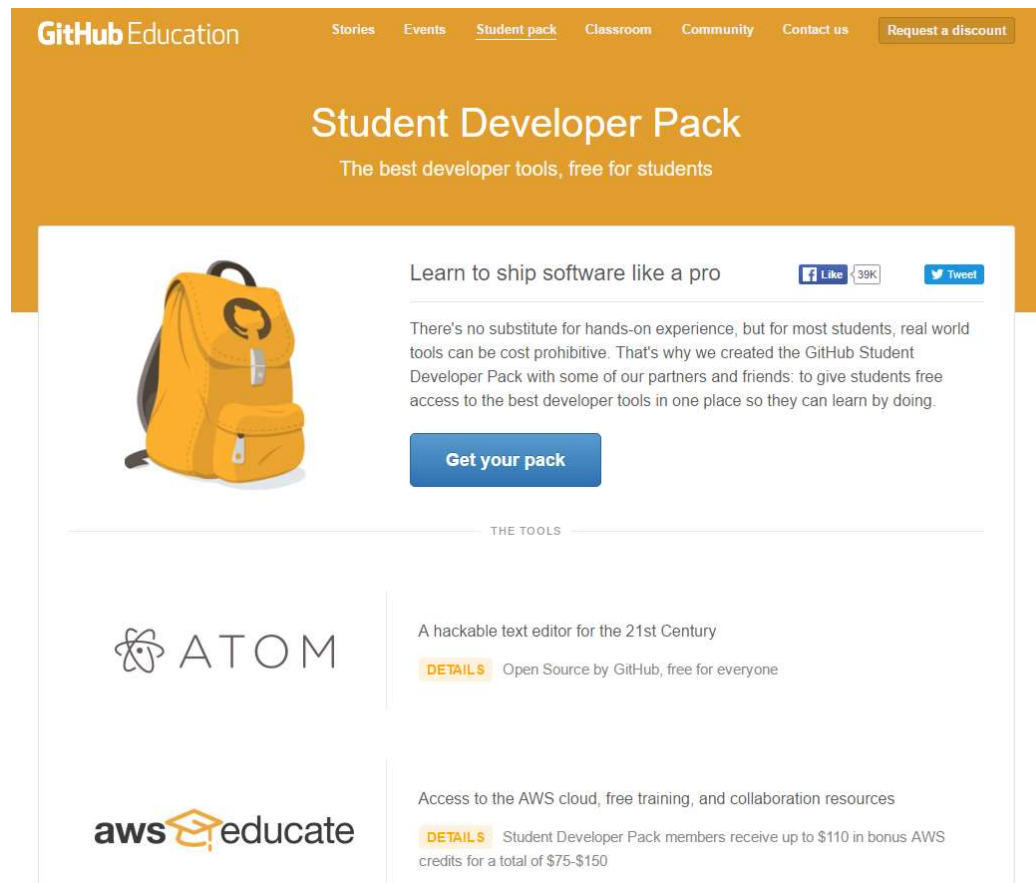
Follow Us

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

Altre risorse utili 1/2

GitHub Education Pack:

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



The screenshot shows the GitHub Education website's landing page for the Student Developer Pack. The page has an orange header 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 subtitle 'The best developer tools, free for students'. Below this is a white content area featuring a yellow backpack icon on the left. To the right of the backpack, the text reads 'Learn to ship software like a pro' with social media share buttons for Facebook (Like 39K) and Twitter (Tweet). A paragraph explains that the pack provides free access to developer tools for students. A blue 'Get your pack' button is positioned below the text. Underneath, a section titled 'THE TOOLS' lists two items: 'ATOM', described as a hackable text editor for the 21st Century, with a 'DETAILS' link and the note 'Open Source by GitHub, free for everyone'; and 'aws educate', which provides access to the AWS cloud, free training, and collaboration resources, with a 'DETAILS' link and the note '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/>



The screenshot shows the top navigation bar of the AWS Educate website. It includes a menu icon, the Amazon Web Services logo, and a list of navigation links: Products, Solutions, Pricing, Software, Support, Customers, Partners, Enterprises, Startups, and Public Sector. On the right side, there are links for English, My Account, and a yellow button labeled 'Sign In to the Console'. Below the navigation bar is a blue banner with white text that reads: 'With the increasing demand for cloud employees, AWS Educate provides an academic gateway for the next generation of IT and cloud professionals. AWS Educate is Amazon's global initiative to provide students and educators with the resources needed to accelerate cloud-related learning endeavors.' A yellow button labeled 'Join AWS Educate Today' is centered in the banner.



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 screenshot shows the Kaggle homepage with a dark header. The main content area is titled "Welcome to Kaggle Competitions" and includes three columns of instructions: "New to Data Science?", "Build a Model", and "Make a Submission". Below this is a list of active competitions, with the top two being "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

Active	All	Entered	All Categories	Sort By	Prize
	Data Science Bowl 2017	Can you improve lung cancer detection? <i>Featured</i> · 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? <i>Featured</i> · A month to go · 326 kernels			\$150,000 1,734 teams

The screenshot shows the CrowdANALYTIX homepage with a dark header. The main content area is titled "Welcome" and includes a three-step process: "1. JOIN", "2. SUBMIT", and "3. EARN". Below this is a search bar and a list of featured 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](#)

Search here ...

Featured Open Closed All
Modeling Research Visualization

Featured Competitions

- Research** (CLOSED)
- Visualization** (CLOSED)
- Modeling** (CLOSED)

Corsi online

- 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/>

Python - Introduzione

- Python é un linguaggio di programmazione dinamico e interpretato
- Non ha un type system statico per dichiarare variabili, parametri e funzioni nel codice (lo ha dinamico, eseguito a runtime)
- Non si perde tempo in compilazione ...
- ... ma gli errori si manifestano in fase di esecuzione

Elementi principali

- Programma o **script**: sequenza di definizioni e comandi
- **Shell**: interpreti di script o comandi;
- **Shell prompt** “>>>”
- Comandi o “**statement**”: istruzioni per l’interprete
- **Oggetti scalari** (int, float, bool, None) e **non scalari** (e.g., stringhe); “**type**” di un oggetto
- **Oggetti** e **operatori** vengono usati per scrivere **espressioni** che vengono valutati in altri oggetti

Operatori per int e float

- Operatori aritmetici:

- $i+j$, $i-j$, $i*j$, i/j
- $i // j$ divisione intera
- $i \% j$ resto della divisione intera
- $i**j$ i elevato alla j

- Operatori di confronto

- $==$ $>$ $>=$ $<=$ $<$
- $!=$ diverso

Variabili e assegnamenti

- **Variabile:** offre un modo per associare un nome a ad un oggetto

```
>>> pi = 3.1415
```

```
>>> msg = "ritenta, sarai più fortunato"
```

```
>>> flag = (13 < 11)
```

- **Assegnamento:** assegnazione di un nome ad un oggetto. Lo stesso oggetto può avere diversi nomi.

- **Assegnamento multiplo:**

```
>>> a, b = 2, 3
```

```
>>> a, b = b, a
```

Parole riservate o Keywords

```
>>> import keyword
>>> print(keyword.kwlist)
['False', 'None', 'True', 'and', 'as',
 'assert', 'break', 'class', 'continue',
 'def', 'del', 'elif', 'else', 'except',
 'finally', 'for', 'from', 'global',
 'if', 'import', 'in', 'is', 'lambda',
 'nonlocal', 'not', 'or', 'pass',
 'raise', 'return', 'try', 'while',
 'with', 'yield']
>>> keyword.iskeyword("bella li")
```

Tipi di errori

- **ValueError**: invalid literal for int() with base 10: '3.0'
- **AttributeError**: readonly attribute (`z.real = 3`)
- **NameError**: name 'x' is not defined (dopo `del x`)
- **TypeError**: unsupported operand type(s) for `**` or `pow()`: 'str' and 'int'
- **TypeError**: 'int' object is not iterable