IPython Notebook

Finally a scalable solution for Coding in the browser

Omoju Miller
@omojumiller

Why is this even important?


By bringing the next billion of our colleagues CS Education

The Beauty and Joy of Computing (CS Principles), Part 2

A computer science principles course intended to broaden participation in computing to non-traditional groups. Part 2 of 4.

About this Course

The Beauty and Joy of Computing (BJC) is a computer science principles course developed at the University of California, Berkeley, intended to broaden participation in computing to non-traditional groups. Computing has profoundly changed the world, opening up wonderful new ways for people to connect, design, research, play, create, and express themselves. However, just *using* a computer is only a small part of the picture. The real transformative and empowering experience comes when one learns how to *program* the computer, to translate ideas into code. This course teaches students how to do exactly that, using Snap*!* (based on Scratch), one of the friendliest programming languages ever invented. It's purely graphical, which means programming involves simply dragging blocks around, and building bigger blocks out of smaller blocks. But this course is far more than just learning to program. We focus on seven big ideas (creativity, abstraction, data and information, algorithms, programming, the Internet, and global impact), and six computational thinking practices (connecting computing, creating computational artifacts, abstracting, analyzing problems and artifacts, communicating, and collaborating). Throughout the course, relevance is emphasized: relevance to the student and to society. This fun, introductory course is not just for computer science majors, it's for everyone... join us!

Berkeley
UNIVERSITY OF CALIFORNIA

BerkeleyX

School:	UC BerkeleyX
Course Code:	BJC.2x
Classes Start:	26 Oct 2015
Course Length:	7 weeks

Prerequisites:

None.

Enroll in **BJC.2x**
and choose your student track

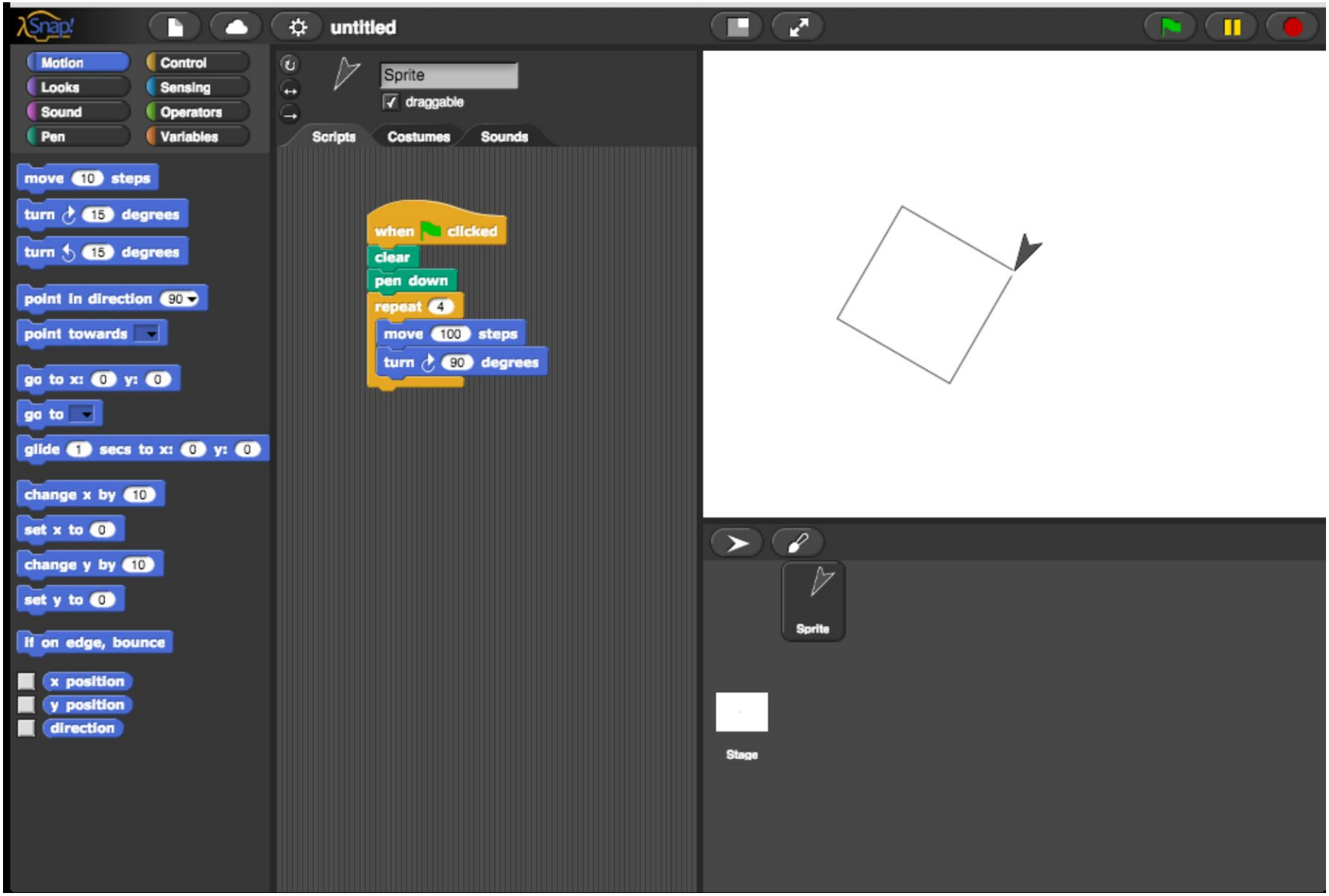
I would like to receive email from The University of California Berkeley and learn about its other programs

31

Student Reviews

0 Reviews

Write a review


Snap!
untitled
Motion
Control
Looks
Sensing
Sound
Operators
Pen
Variables
Sprite
draggable
Scripts
Costumes
Sounds
move 10 steps
turn 15 degrees
turn 15 degrees
point in direction 90
point towards
go to x: 0 y: 0
go to
glide 1 secs to x: 0 y: 0
change x by 10
set x to 0
change y by 10
set y to 0
if on edge, bounce
x position
y position
direction
when clicked
clear
pen down
repeat 4
move 100 steps
turn 90 degrees
Sprite
Stage

Data Exploration Module

```
def lexical_diversity(my_text_data):
...     word_count = len(my_text_data)
...     vocab_size = len(set(my_text_data))
...     diversity_score = word_count / vocab_size
...     return diversity_score
```

```
emma = nltk.corpus.gutenberg.words('austen-emma.txt')

print "The lexical diversity score for the first 35,000 words in the Jay Z corpus is ", lexical_diversity(the_corpus
[:35000])
print "The lexical diversity score for the first 35,000 words in the Emma corpus is ", lexical_diversity(emma[:35000
])

#print len(emma)
#print len(set(emma[:35000]))
#print len(set(the_corpus[:3500]))
```

```
The lexical diversity score for the first 35,000 words in the Jay Z corpus is  6
The lexical diversity score for the first 35,000 words in the Emma corpus is  10
```

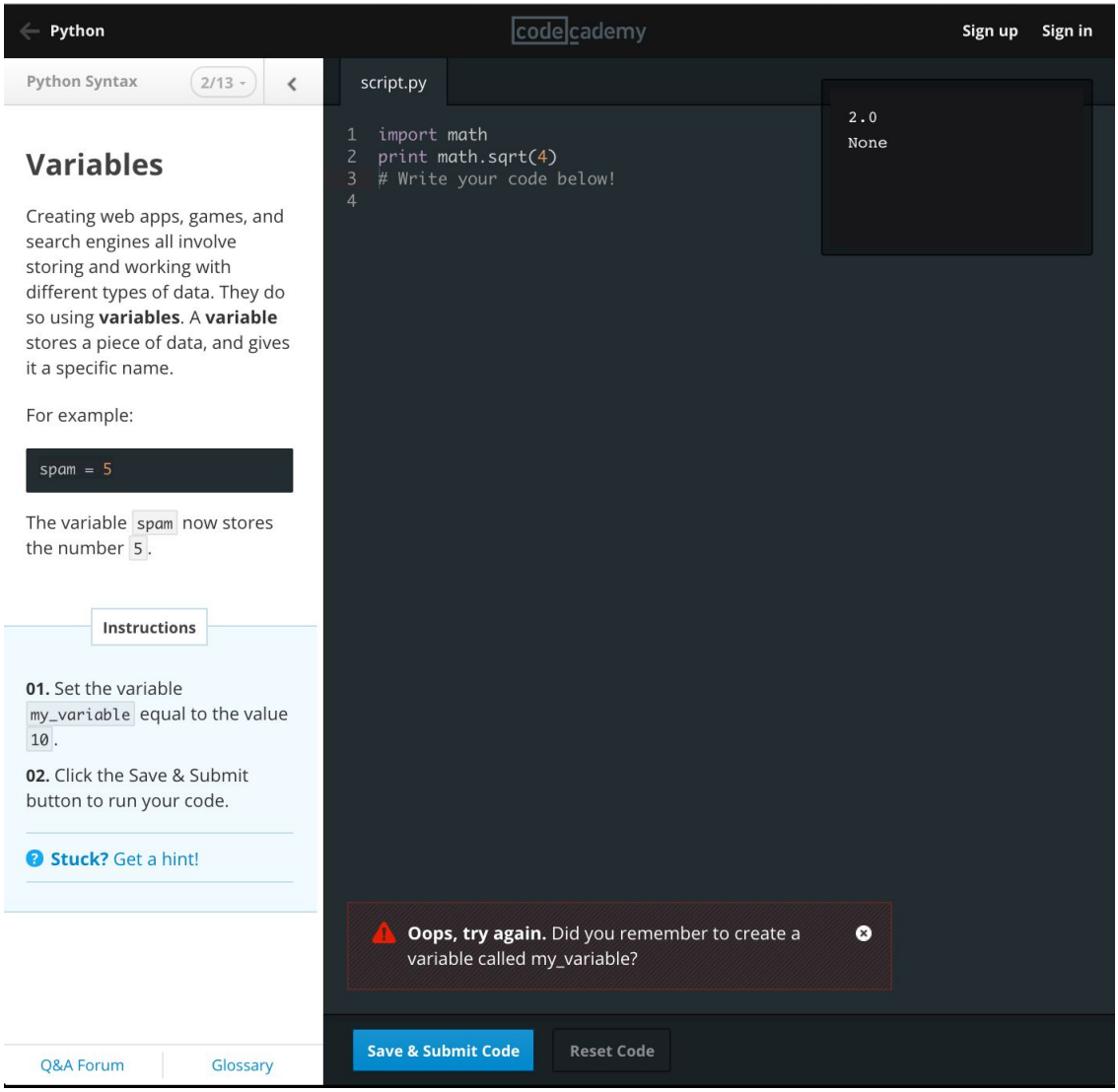
Easy Peasy
Lemon
Squeezy

What was the challenge?

- Build CS courses online, with **EVERYTHING** in the browser
- Make the onboarding experience into writing code as simple as possible
- Powerful computation environment to recreate academic experiments

There are solutions out there

- Skulpt
- Codecademy Editor
- Python Fiddle Editor
- Repl.it Editor
- Runable



All things aren’t equal

- Cloud9
- IPython - Notebook

Powerful **Workspaces**

Set up your system without any hassle

Simply pick your configuration and develop your app. No need to spend valuable development time on system setup and maintenance. You can create, build and run any development stack in seconds. We maintain it, you control it.

All the freedom you'd expect

Workspaces are powered by Docker Ubuntu containers that give you full freedom over your environment, including sudo rights. Do a git push, compile SASS, see server output, and Run apps easily with the built-in Terminal and Runners.

```
sudo apt-get install imagemagick
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Or connect Cloud9 to your own VM via SSH

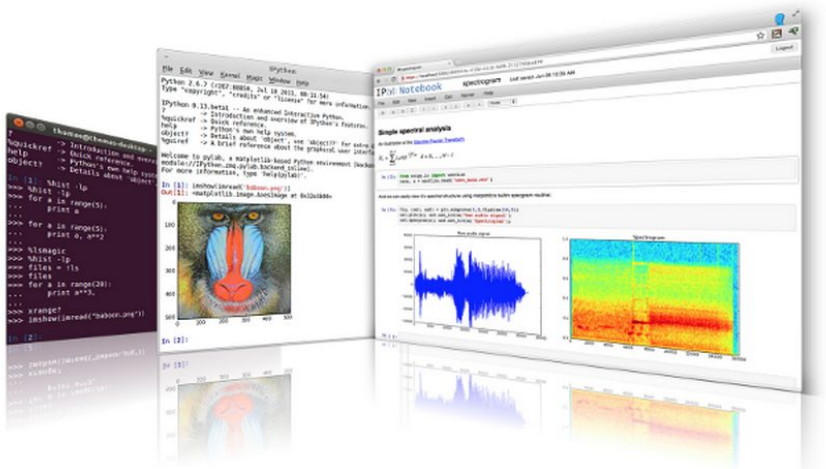
Never lose a line of code

IP[y]: IPython
Interactive Computing

Install · **Docs** · **Videos** · **News** · **Cite** · **Sponsors** · **Donate**

IPython provides a rich architecture for interactive computing with:

- Powerful interactive shells (terminal and Qt-based).
- A browser-based notebook with support for code, text, mathematical expressions, inline plots and other rich media.
- Support for interactive data visualization and use of GUI toolkits.
- Flexible, embeddable interpreters to load into your own projects.
- Easy to use, high performance tools for parallel computing.



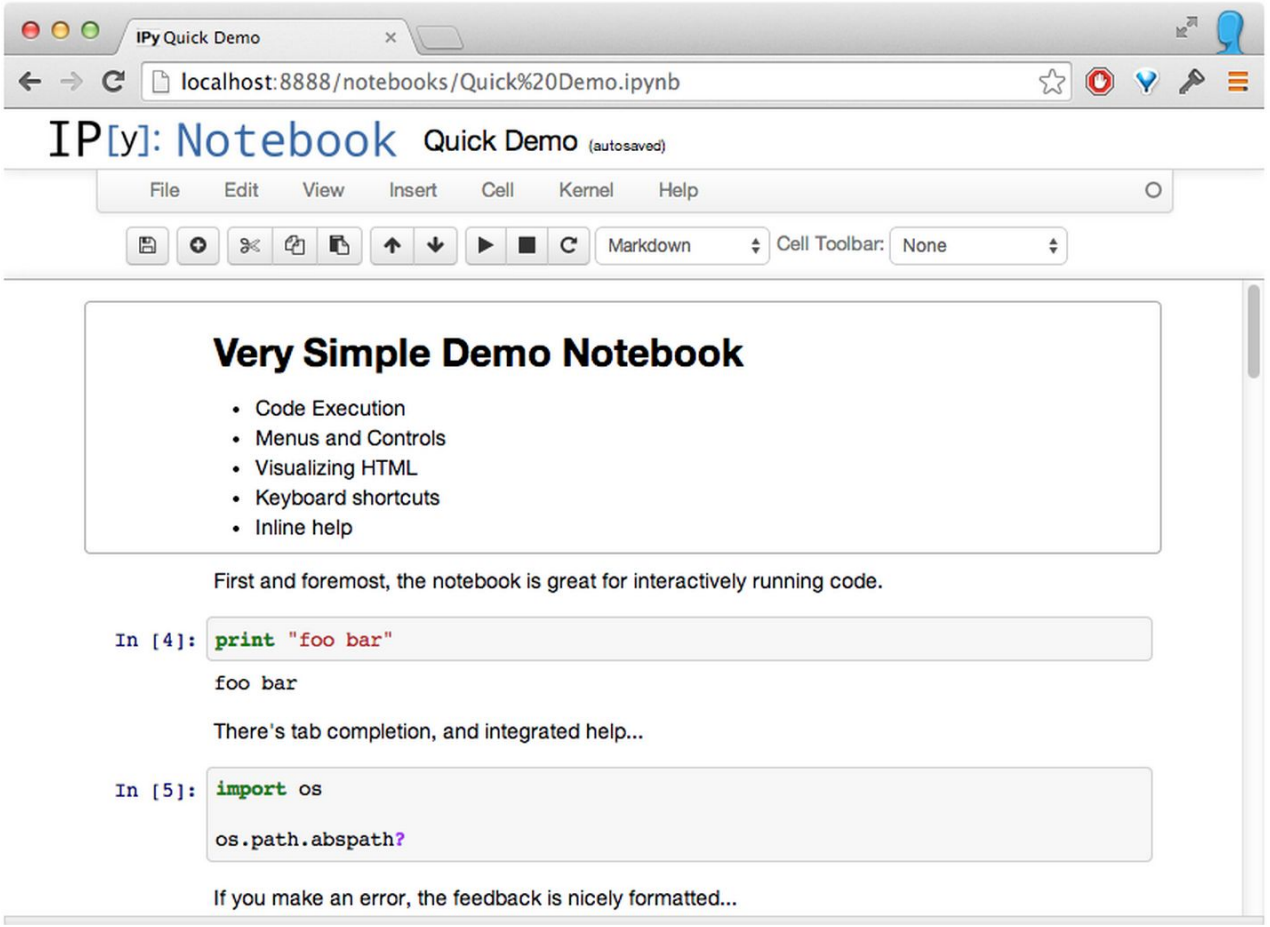
Narrative is King

”literate computing”: it is the weaving of a narrative directly into a live computation, interleaving text with code.

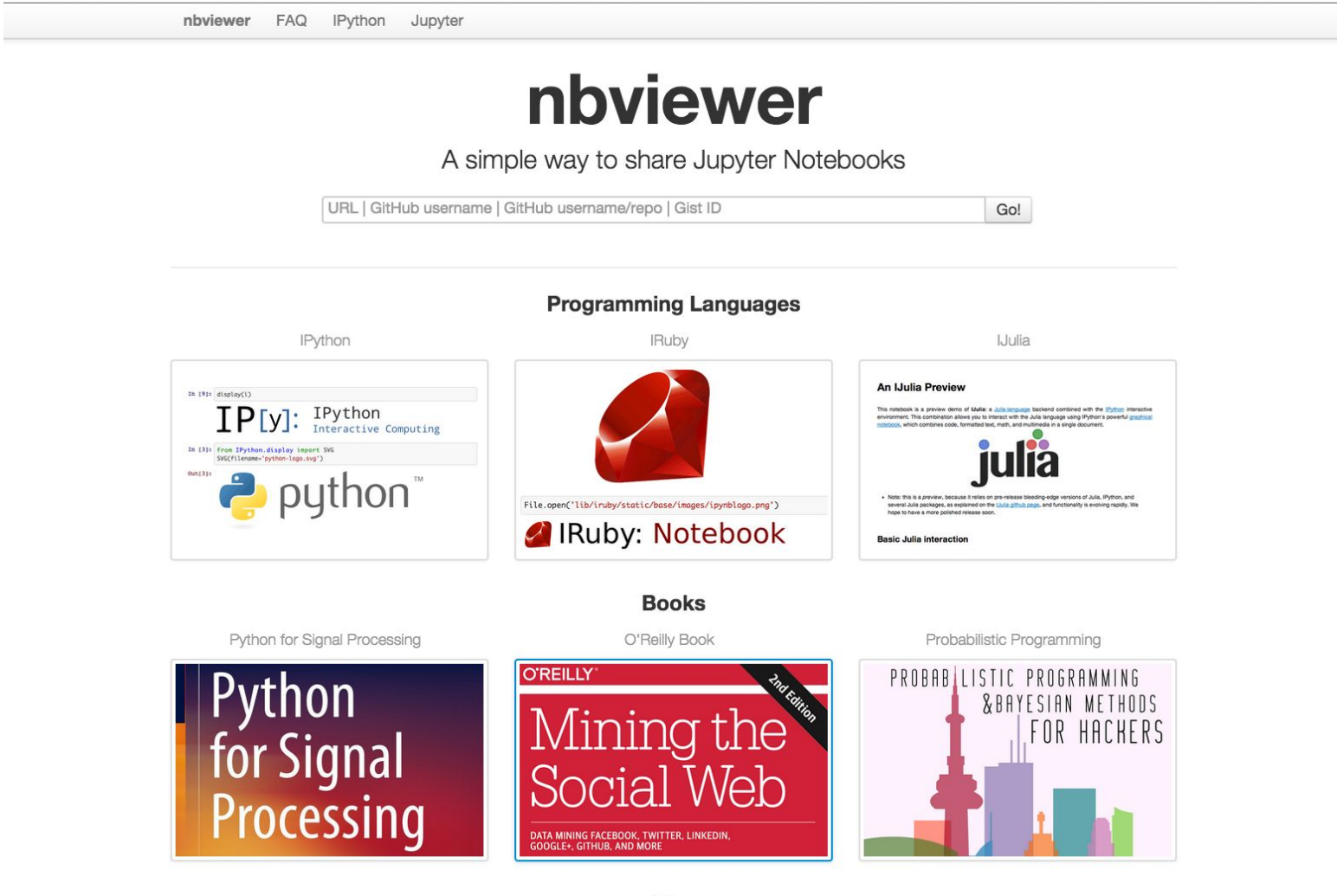
- Fernando Perez

What is the notebook?

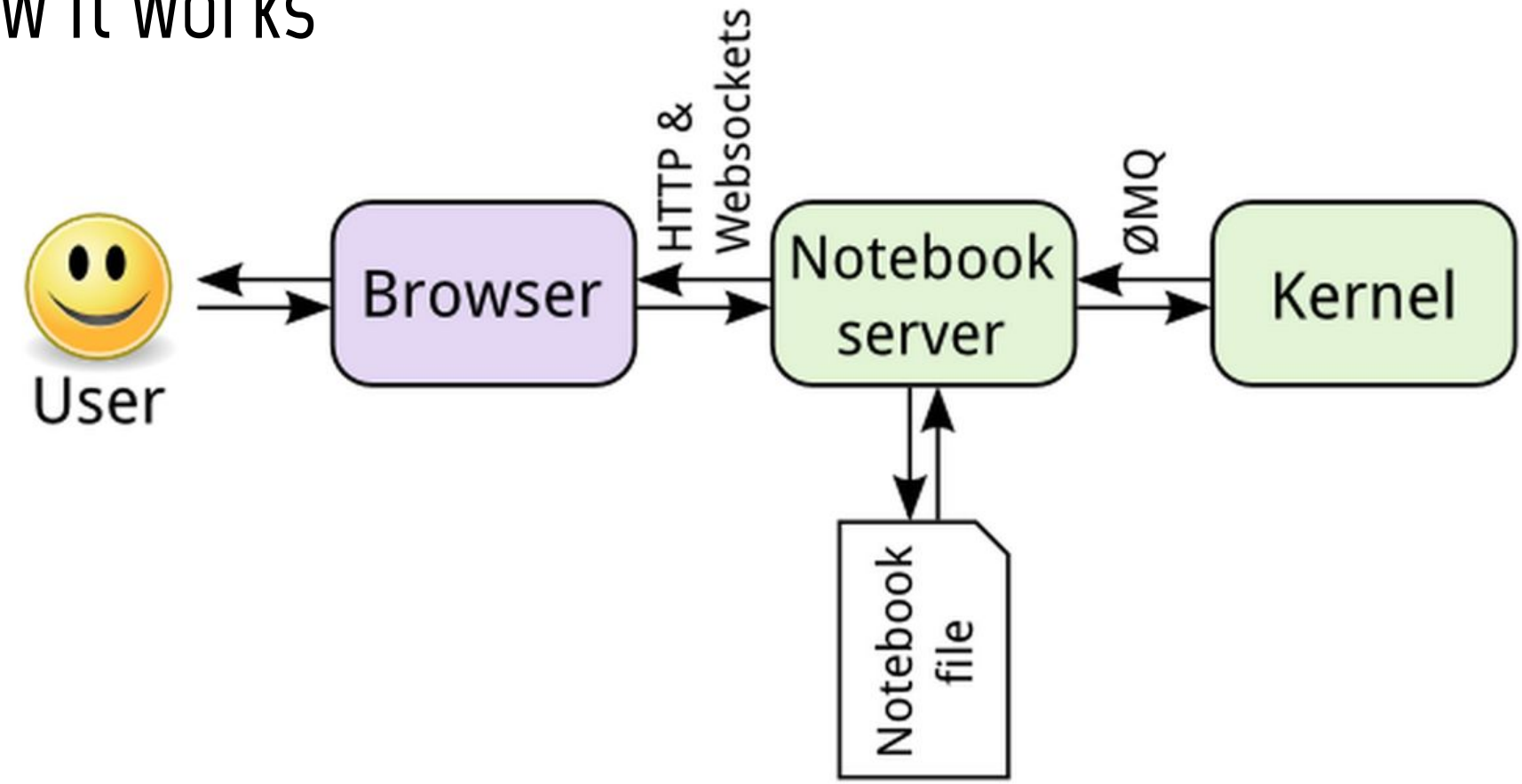
A ”browser-based interactive computing environment”

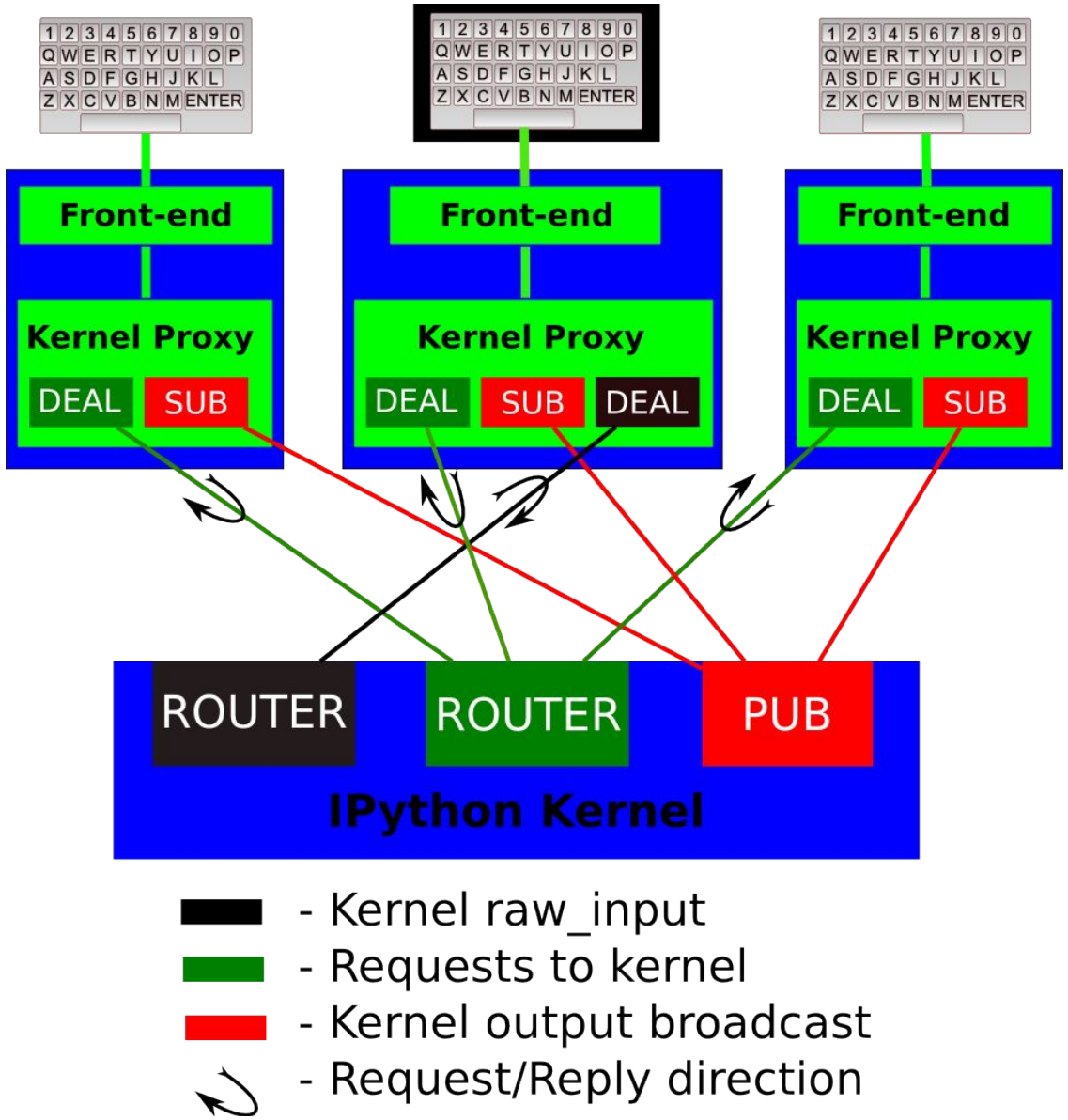


NBViewer


nbviewer
FAQ
IPython
Jupyter
nbviewer
A simple way to share Jupyter Notebooks
URL | GitHub username | GitHub username/repo | Gist ID
Go!
Programming Languages
IPython
IRuby
IJulia
IPython
Interactive Computing
python
IRuby: Notebook
An IJulia Preview
julia
Basic Julia interaction
Books
Python for Signal Processing
O'Reilly Book
Probabilistic Programming
Python for Signal Processing
O'REILLY
2nd Edition
Mining the Social Web
DATA MINING FACEBOOK, TWITTER, LINKEDIN, GOOGLE+, GITHUB, AND MORE
PROBABILISTIC PROGRAMMING &BAYESIAN METHODS FOR HACKERS

How it works


User
Browser
HTTP &
Websockets
Notebook
server
ØMQ
Kernel
Notebook
file


Front-end
Kernel Proxy
DEAL
SUB
Front-end
Kernel Proxy
DEAL
SUB
DEAL
Front-end
Kernel Proxy
DEAL
SUB
ROUTER
ROUTER
PUB
IPython Kernel
- Kernel raw_input
- Requests to kernel
- Kernel output broadcast
- Request/Reply direction

Other languages

- Ruby
- Scala
- Julia
- Cython

Installing IPython

There are multiple ways of installing IPython. This page contains simplified installation instructions that should work for most users. Our official documentation contains detailed instructions for manual installation targeted at advanced users and developers.

I already have Python

If you already have Python installed and are familiar with installing packages, you can get IPython with **pip**:

```
pip install ipython
```

Or if you want to also get the dependencies for the IPython notebook:

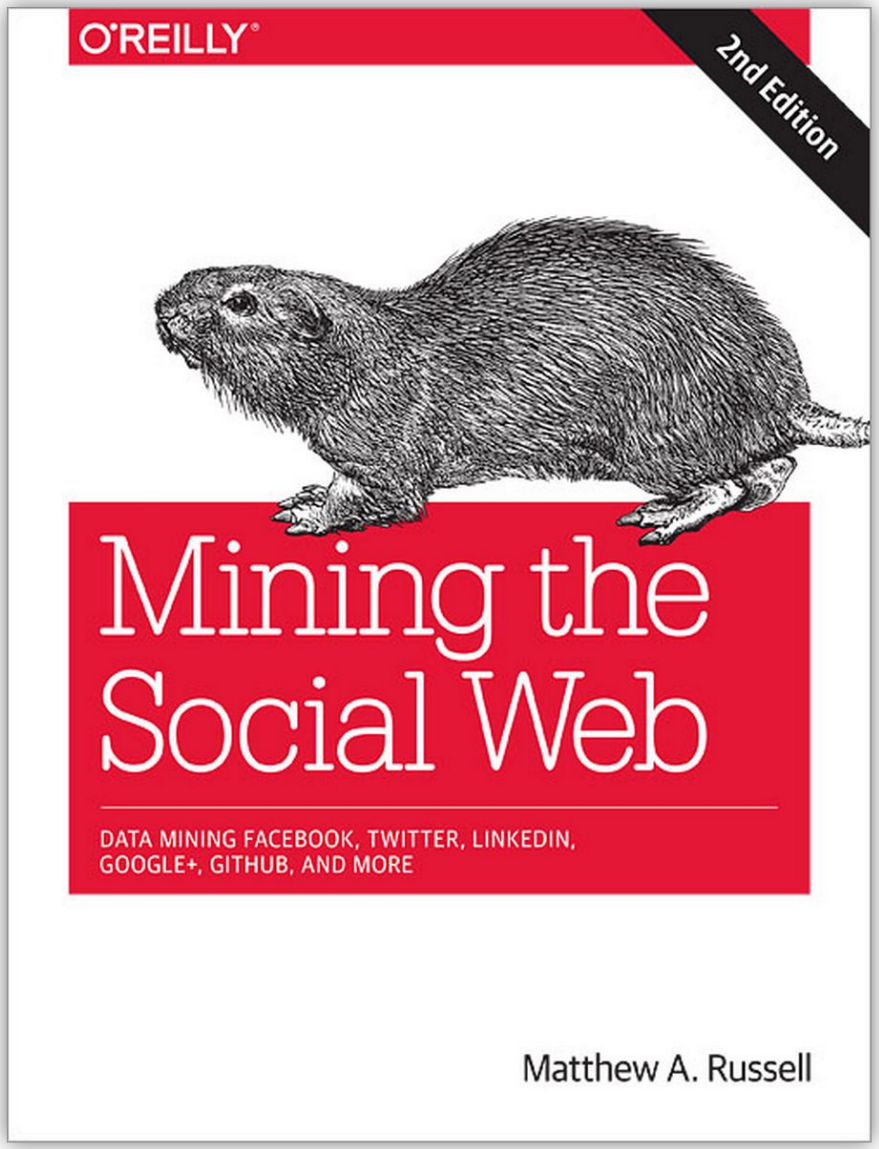
```
pip install "ipython[notebook]"
```

I am getting started with Python

For new users who want to install a full Python environment for scientific computing and data science, we suggest installing the Anaconda or Canopy Python distributions, which provide Python 2.7, IPython and all of its dependences as well as a complete set of open source packages for scientific computing and data science.

1. Download and install Continuum's Anaconda or the free edition of Enthought's Canopy.
2. Update IPython to the current version using the Terminal:

This book's examples are written entirely using the IPython Notebook



ACADEMY
OF MOTION PICTURE
ARTS AND SCIENCES

Results for **#oscars** Save

Top / All

E! Online @eonline · 16m

The best and worst moments from the 2015 #**Oscars**: eonli.ne/1GjwrNS

76 212

View photo

Photos · View all

Example 1. Authorizing an application to access Twitter account data

[2]:
```
import twitter

# XXX: Go to http://dev.twitter.com/apps/new to create an app and get values
# for these credentials, which you'll need to provide in place of these
# empty string values that are defined as placeholders.
# See https://dev.twitter.com/docs/auth/oauth for more information
# on Twitter's OAuth implementation.

CONSUMER_KEY = 'fQHr2IXQQ6RKf7Ev38fWxuHBA'
CONSUMER_SECRET ='g0vNMJS5ZGw7GQ87WozwebettgAQywhzlJRP5h8atQrVY9iRtU'
OAUTH_TOKEN = '303569935-L0V3tn0AR5dzAA0NUu4hf5hojpeNVRltFrDTvHR8'
OAUTH_TOKEN_SECRET = 'uEwCiASeAo3PVp45K8tYFD9T0g6Sv4STMnrHj6bmQ9CJt'

auth = twitter.oauth.OAuth(OAUTH_TOKEN, OAUTH_TOKEN_SECRET,
                           CONSUMER_KEY, CONSUMER_SECRET)

twitter_api = twitter.Twitter(auth=auth)

# Nothing to see by displaying twitter_api except that it's now a
# defined variable

print twitter_api
```

```
<twitter.api.Twitter object at 0x10490b750>
```

Example 6. Extracting text, screen names, and hashtags from tweets

In [8]:
```
status_texts = [ status['text']
                 for status in statuses ]

screen_names = [ user_mention['screen_name']
                 for status in statuses
                     for user_mention in status['entities']['user_mentions'] ]

hashtags = [ hashtag['text']
             for status in statuses
                 for hashtag in status['entities']['hashtags'] ]

# Compute a collection of all words from all tweets
words = [ w
          for t in status_texts
              for w in t.split() ]

# Explore the first 5 items for each...

print json.dumps(status_texts[0:5], indent=1)
print json.dumps(screen_names[0:5], indent=1)
print json.dumps(hashtags[0:5], indent=1)
print json.dumps(words[0:5], indent=1)
```

```
[
 "RT @casacelis: Ever dreamed on get an Oscar statue? Now you can build your #Lego one. #Oscars #Oscars2015 #LEGOFriends #LegoF
an #Fan http:/\u2026",
 "RT @TheTimMcGraw: This is what I walked in my home to this morn!! #Oscars http://t.co/VUL69sBrOX",
 "RT @justinbieber: Good luck to everyone tonight. Thanks for all the great movies. #Oscars",
 "RT @justinbieber: Good luck to everyone tonight. Thanks for all the great movies. #Oscars",
 "RT @justinbieber: Good luck to everyone tonight. Thanks for all the great movies. #Oscars"
]
[
 "casacelis",
 "TheTimMcGraw",
 "justinbieber",
 "justinbieber",
 "justinbieber"
```

IPython Magic: Development Powertools

Which method is faster?

In [14]:
```
import random, string

# make a big list of random strings
words = [''.join(random.choice(string.ascii_uppercase) for _ in range(6)) for _ in rang
e(1000)]

# Plan A: turn them all into lowercase with a list comprehension
def listcomp_lower(words):
    return [w.lower() for w in words]

# Plan B: Start with a list, and word by word append the lowercase versions
def append_lower(words):
    new = []
    for w in words:
        new.append(w.lower())
    return new

# %timeit is IPython Magic to do a quick benchmark
%timeit append_lower(words)
```

```
1000 loops, best of 3: 249 µs per loop
```

In [15]:
```
%timeit listcomp_lower(words)
```

```
10000 loops, best of 3: 179 µs per loop
```

Use Cases

Code Mentoring

Teaching

Data Analysis

Writing Books

Teaching

The divide-and-conquer strategy solves a problem by:

1. Breaking it into subproblems that are themselves smaller instances of the same type of problem
2. Recursively solving these subproblems
3. Appropriately combining their answers

The real work is done piecemeal, in three different places: in the partitioning of problems into subproblems; at the very tail end of the recursion, when the subproblems are so small that they are solved outright; and in the gluing together of partial answers. These are held together and coordinated by the algorithm's core recursive structure. As an introductory example, we'll see how this technique yields a new algorithm for multiplying numbers, one that is much more efficient than the method we all learned in elementary school!

More on Runtimes

1. $logN$: Commonly occurs in programs that solve big problems, cutting down the problem size by some constant fraction at each step. Whenever N doubles, $logN$ increases by a constant, but $logN$ does not double until N increases to N^2
2. $NlogN$: This runtime usually occurs when we break up a problem into smaller subproblems, and somehow recombine them to find the solution. When N is 1,000,000, $NlogN$ is around 20,000,000. When N doubles, running time doubles, but not by much

lgN	$\sqrt{N}$	N	$NlgN$	$NlgN^2$	$N^{\frac{3}{2}}$	N^2
3	3	10	33	110	32	100
7	10	100	664	4410	1000	10000
10	32	1000	9966	993317	31623	1000000
13	100	10000	132877	1765633	1000000	100000000
17	316	100000	1660964	27588016	31622777	10000000000

In [1]:
```
N = 100000
print N * N
```

```
10000000000
```

In [4]:
```
import math
def log2( x ):
    return math.log( x ) / math.log( 2 )
```

In [5]:
```
log2(10000)
```

Out[5]:
```
13.28771237954945
```

In [6]:
```
from IPython.display import YouTubeVideo
# a talk about Measuring program efficiency and algorithm complexity
# Video credit: John Denero
YouTubeVideo('DWhsOSHjY98?list=PL6BsET-8jgYVQtu1EXVXv_pYyjfj5r-EA')
```

Out[6]:

Analysis

This is the analysis of the data that was collected in Fall 2014 from CS10 and CS61A

In [446]:
```
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import numpy as np
```

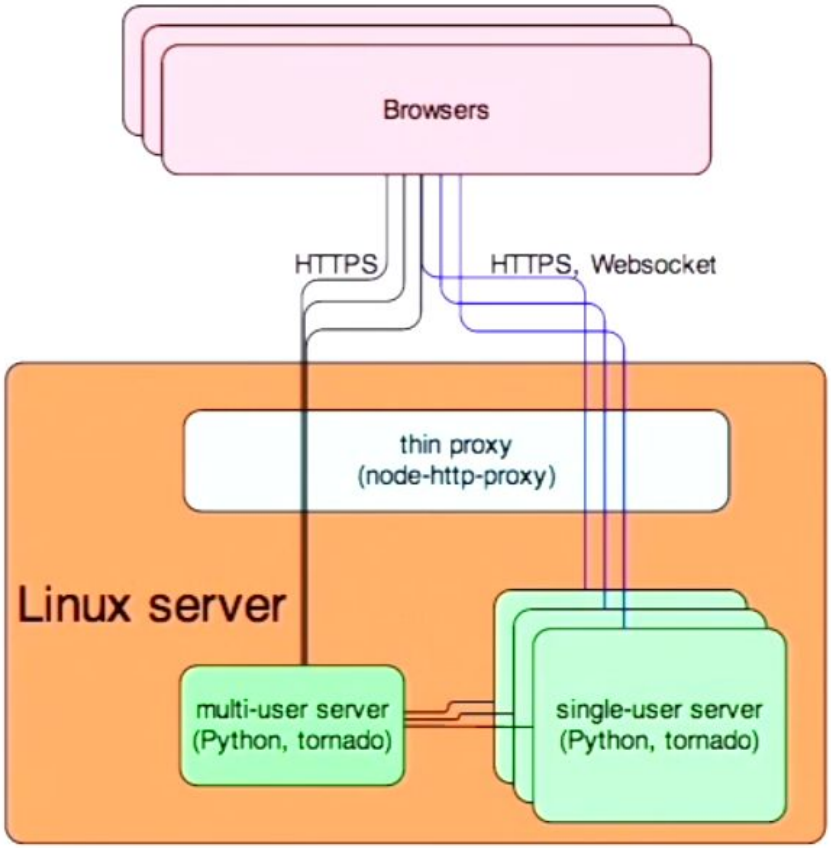
In [447]:
```
from pandas import DataFrame
# (*) Pandas for data manipulation
import pandas as pd

# Read csv file and define dataframe object
cs10_df = pd.read_csv('CS10_Pre_Responses_Fall2014.csv')
cs10_df.head()  # show dataframe header to stdout
```

Out[447]:

	Timestamp	CONSENT If you agree to participate, please do so by clicking the "I agree" button on this survey.	Could I please know your gender	What is your reason for taking this class?	Do you have a declared major?	I like to use Computer Science to solve problems.	Knowledge of computing will allow me to secure a good job.	I can learn to understand computing concepts.	My career goals do not require that I learn computing skills.	Women are less capable of success in CS than men.	...	Our class materials (e.g., case studies and projects) were relevant and practical.	Did you take a CS course in High School?	Did you have any exposure to Computer Science before UC Berkeley?	Did a family memb introd you to Comp Scien
0	11/7/2014 11:08:40	I agree	Female	Interested	Applied Math	3	4	4	2	1	...	3	No	No	No
1	11/11/2014 15:58:53	I agree	Female	Interested	Business	1	1	1	3	1	...	3	No	No	Yes
2	11/5/2014 18:12:15	I agree	Female	Interested	Chemical Biology	5	5	5	3	1	...	4	No	No	No
3	11/6/2014 16:47:48	I agree	Female	Interested	Chemistry	5	5	4	2	1	...	4	No	No	No
4	11/6/2014 12:35:41	I agree	Female	Interested	Cognitive Science	3	5	3	2	1	...	4	No	No	No

Multi-user notebook server



Live Collaboration

Integration into GoogleDrive Classroom