Python

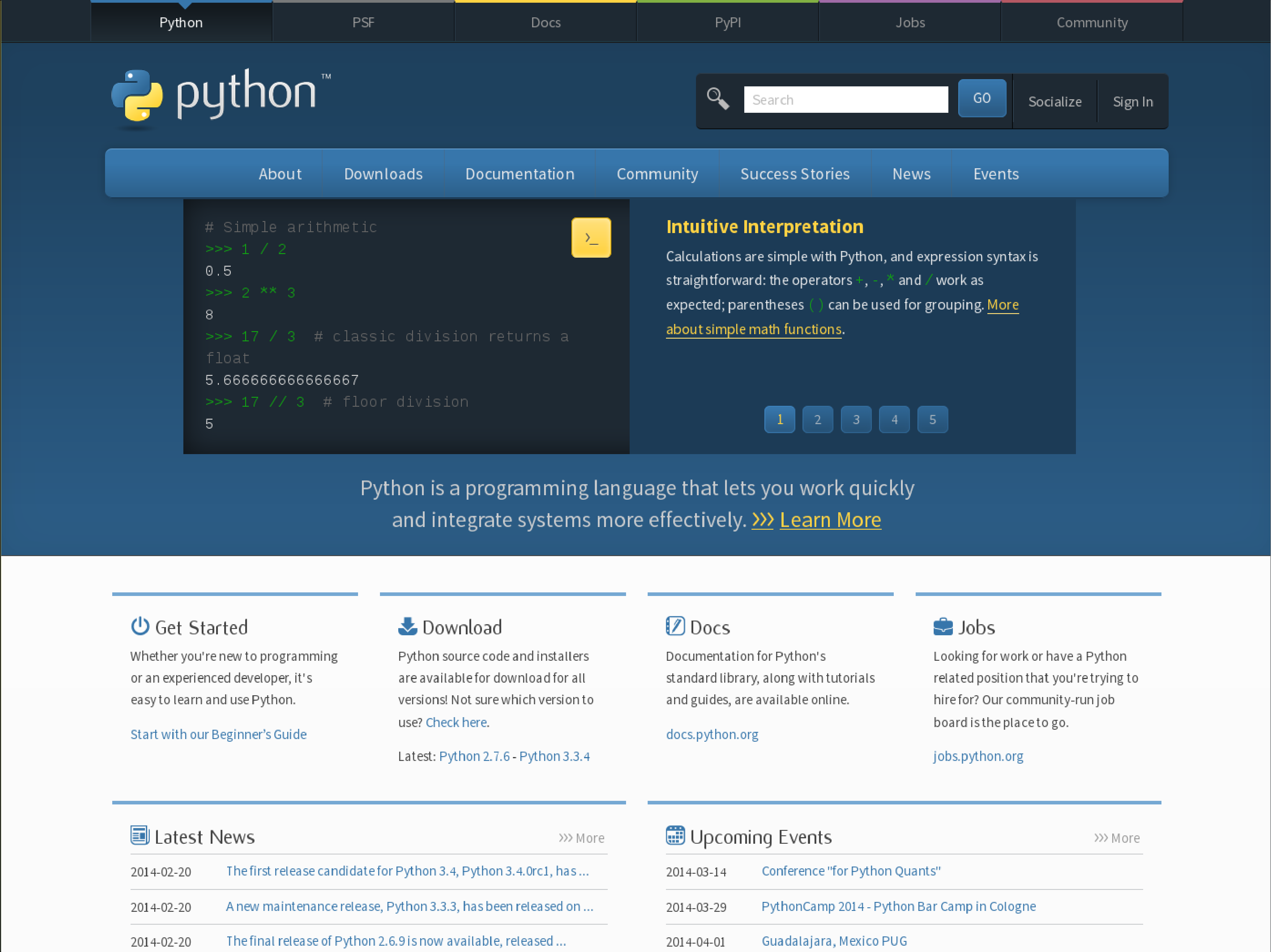
the awesome programming language

Python

- created by Guido van Rossum
- interpreted, interactive, object-oriented
- easy to learn and powerful

start here

http://www.python.org/



```
# Simple arithmetic
>>> 1 / 2
0.5
>>> 2 ** 3
8
>>> 17 / 3  # classic division returns a
float
5.666666666666667
>>> 17 // 3  # floor division
5
```

Intuitive Interpretation

Calculations are simple with Python, and expression syntax is straightforward: the operators +, -, * and / work as expected; parentheses ( ) can be used for grouping. More about simple math functions.

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.

Start with our Beginner's Guide

Download

Python source code and installers are available for download for all versions! Not sure which version to use? Check here.

Latest: Python 2.7.6 - Python 3.3.4

Docs

Documentation for Python's standard library, along with tutorials and guides, are available online.

docs.python.org

Jobs

Looking for work or have a Python related position that you're trying to hire for? Our community-run job board is the place to go.

jobs.python.org

Latest News

>>> More

2014-02-20 The first release candidate for Python 3.4, Python 3.4.0rc1, has ...

2014-02-20 A new maintenance release, Python 3.3.3, has been released on ...

2014-02-20 The final release of Python 2.6.9 is now available, released ...

Upcoming Events

>>> More

2014-03-14 Conference "for Python Quants"

2014-03-29 PythonCamp 2014 - Python Bar Camp in Cologne

2014-04-01 Guadalajara, Mexico PUG

community

documentation

http://www.python.org/doc/

Download

Download these documents

Docs for other versions

Python 3.3 (stable)
Python 3.4 (in development)
Old versions

Other resources

PEP Index
Beginner's Guide
Book List
Audio/Visual Talks

Quick search

Go

Enter search terms or a module, class or function name.

Python v2.7.6 documentation

Welcome! This is the documentation for Python 2.7.6, last updated Mar 07, 2014.

Parts of the documentation:

What's new in Python 2.7?
or all "What's new" documents since 2.0

Tutorial
start here

Library Reference
keep this under your pillow

Language Reference
describes syntax and language elements

Python Setup and Usage
how to use Python on different platforms

Python HOWTOs
in-depth documents on specific topics

Extending and Embedding
tutorial for C/C++ programmers

Python/C API
reference for C/C++ programmers

Installing Python Modules
information for installers & sys-admins

Distributing Python Modules
sharing modules with others

FAQs
frequently asked questions (with answers!)

Indices and tables:

Global Module Index
quick access to all modules

General Index
all functions, classes, terms

Glossary
the most important terms explained

Search page
search this documentation

Complete Table of Contents
lists all sections and subsections

Meta information:

Reporting bugs

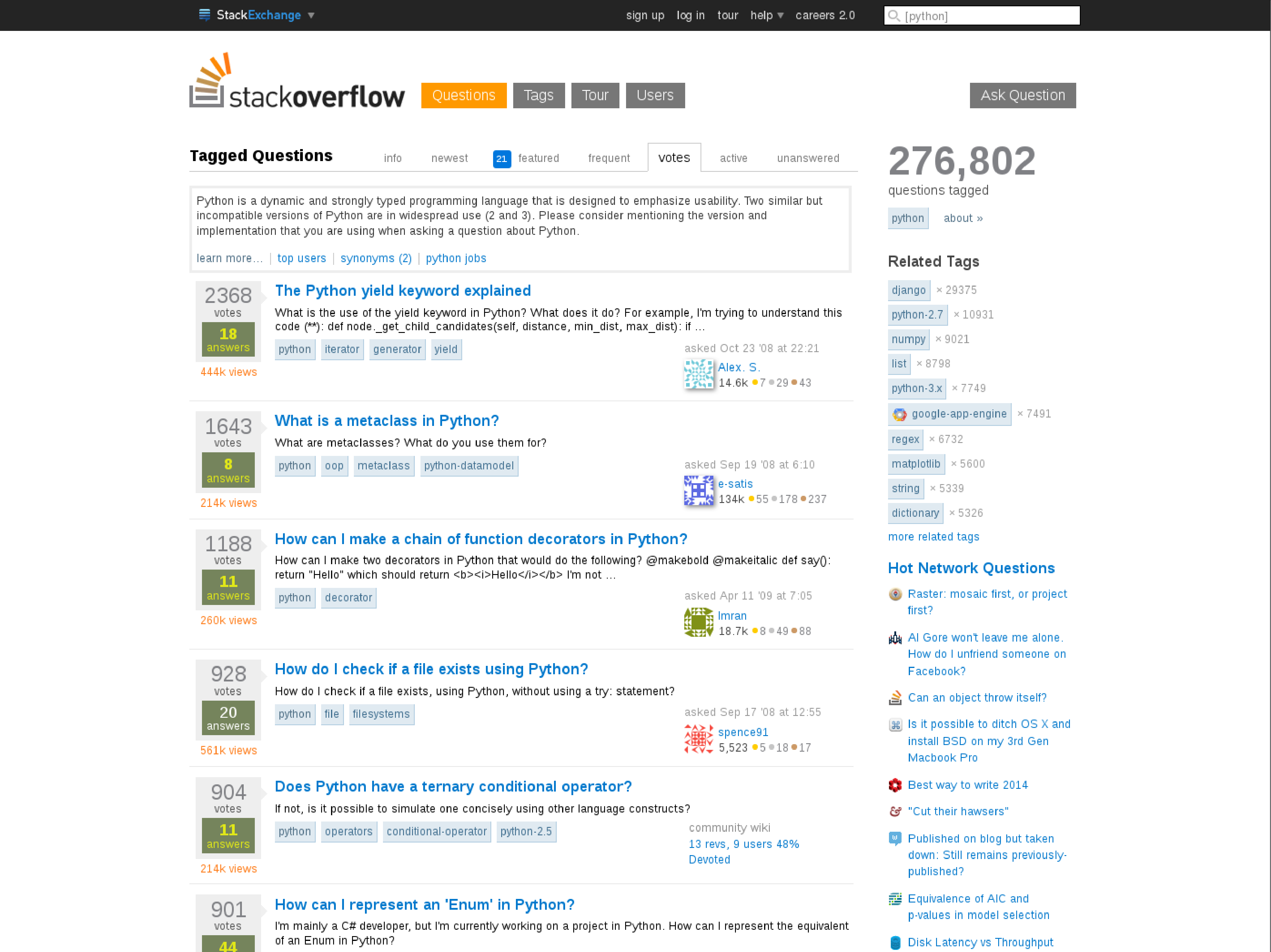
About the documentation

History and License of Python

Copyright

Stack Overflow

http://stackoverflow.com/questions/tagged/python



StackExchange ▾ sign up log in tour help ▾ careers 2.0 [python]

stackoverflow Questions Tags Tour Users Ask Question

Tagged Questions

info newest 21 featured frequent votes active unanswered

Python is a dynamic and strongly typed programming language that is designed to emphasize usability. Two similar but incompatible versions of Python are in widespread use (2 and 3). Please consider mentioning the version and implementation that you are using when asking a question about Python.

learn more… | top users | synonyms (2) | python jobs

2368 votes
18 answers
444k views

The Python yield keyword explained

What is the use of the yield keyword in Python? What does it do? For example, I'm trying to understand this code (**): def node._get_child_candidates(self, distance, min_dist, max_dist): if …

python iterator generator yield

asked Oct 23 '08 at 22:21
Alex. S.
14.6k ●7 ●29 ●43

1643 votes
8 answers
214k views

What is a metaclass in Python?

What are metaclasses? What do you use them for?

python oop metaclass python-datamodel

asked Sep 19 '08 at 6:10
e-satis
134k ●55 ●178 ●237

1188 votes
11 answers
260k views

How can I make a chain of function decorators in Python?

How can I make two decorators in Python that would do the following? @makebold @makeitalic def say(): return "Hello" which should return <b><i>Hello</i></b> I'm not …

python decorator

asked Apr 11 '09 at 7:05
Imran
18.7k ●8 ●49 ●88

928 votes
20 answers
561k views

How do I check if a file exists using Python?

How do I check if a file exists, using Python, without using a try: statement?

python file filesystems

asked Sep 17 '08 at 12:55
spence91
5,523 ●5 ●18 ●17

904 votes
11 answers
214k views

Does Python have a ternary conditional operator?

If not, is it possible to simulate one concisely using other language constructs?

python operators conditional-operator python-2.5

community wiki
13 revs, 9 users 48%
Devoted

901 votes
44

How can I represent an 'Enum' in Python?

I'm mainly a C# developer, but I'm currently working on a project in Python. How can I represent the equivalent of an Enum in Python?

276,802
questions tagged

python about »

Related Tags

django	× 29375
python-2.7	× 10931
numpy	× 9021
list	× 8798
python-3.x	× 7749
google-app-engine	× 7491
regex	× 6732
matplotlib	× 5600
string	× 5339
dictionary	× 5326

more related tags

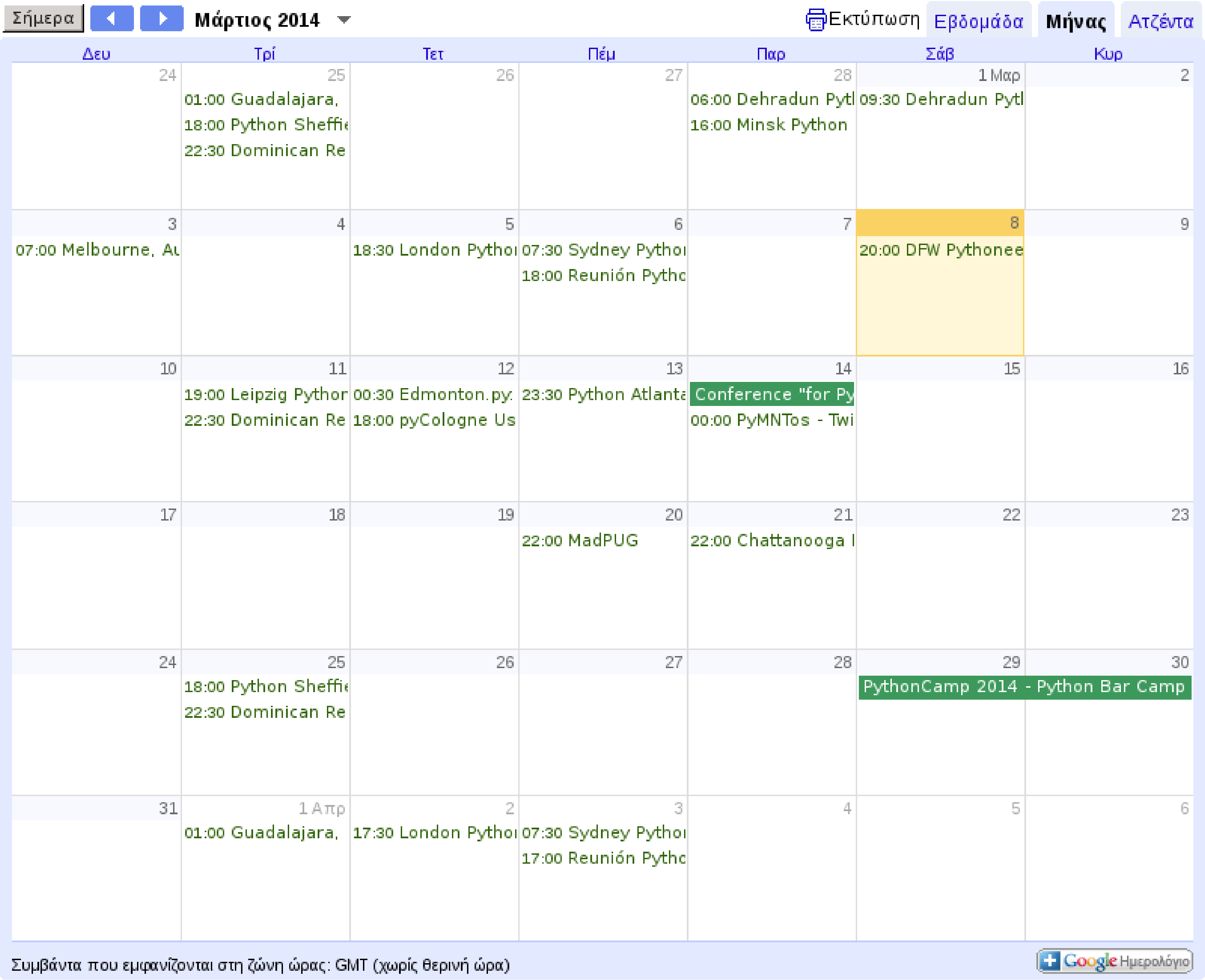
Hot Network Questions

- Raster: mosaic first, or project first?
- Al Gore won't leave me alone. How do I unfriend someone on Facebook?
- Can an object throw itself?
- Is it possible to ditch OS X and install BSD on my 3rd Gen Macbook Pro
- Best way to write 2014
- "Cut their hawsers"
- Published on blog but taken down: Still remains previously-published?
- Equivalence of AIC and p-values in model selection
- Disk Latency vs Throughput

PyCon

http://www.pycon.org/

Python Events and User Group Calendar



Σήμερα ◀ ▶ **Μάρτιος 2014** ▾ Εκτύπωση Εβδομάδα **Μήνας** Ατζέντα

Δευ	Τρί	Τετ	Πέμ	Παρ	Σάβ	Κυρ
24	25 01:00 Guadalajara, 18:00 Python Sheffie 22:30 Dominican Re	26	27	28 06:00 Dehradun Pytl 16:00 Minsk Python	1 Μαρ 09:30 Dehradun Pytl	2
3 07:00 Melbourne, Au	4	5 18:30 London Pythoi	6 07:30 Sydney Pythoi 18:00 Reunión Pytho	7	8 20:00 DFW Pythonee	9
10	11 19:00 Leipzig Pythor 22:30 Dominican Re	12 00:30 Edmonton.py: 18:00 pyCologne Us	13 23:30 Python Atlanta	14 Conference "for Py 00:00 PyMNTos - Twi	15	16
17	18	19	20 22:00 MadPUG	21 22:00 Chattanooga I	22	23
24	25 18:00 Python Sheffie 22:30 Dominican Re	26	27	28	29 PythonCamp 2014 - Python Bar Camp	30
31	1 Απρ 01:00 Guadalajara,	2 17:30 London Pythoi	3 07:30 Sydney Pythoi 17:00 Reunión Pytho	4	5	6

Συμβάντα που εμφανίζονται στη ζώνη ώρας: GMT (χωρίς θερινή ώρα)

Python Software Foundation

"The mission of the Python Software Foundation is to promote, protect, and advance the Python programming language, and to support and facilitate the growth of a diverse and international community of Python programmers."

http://www.python.org/psf/

batteries included

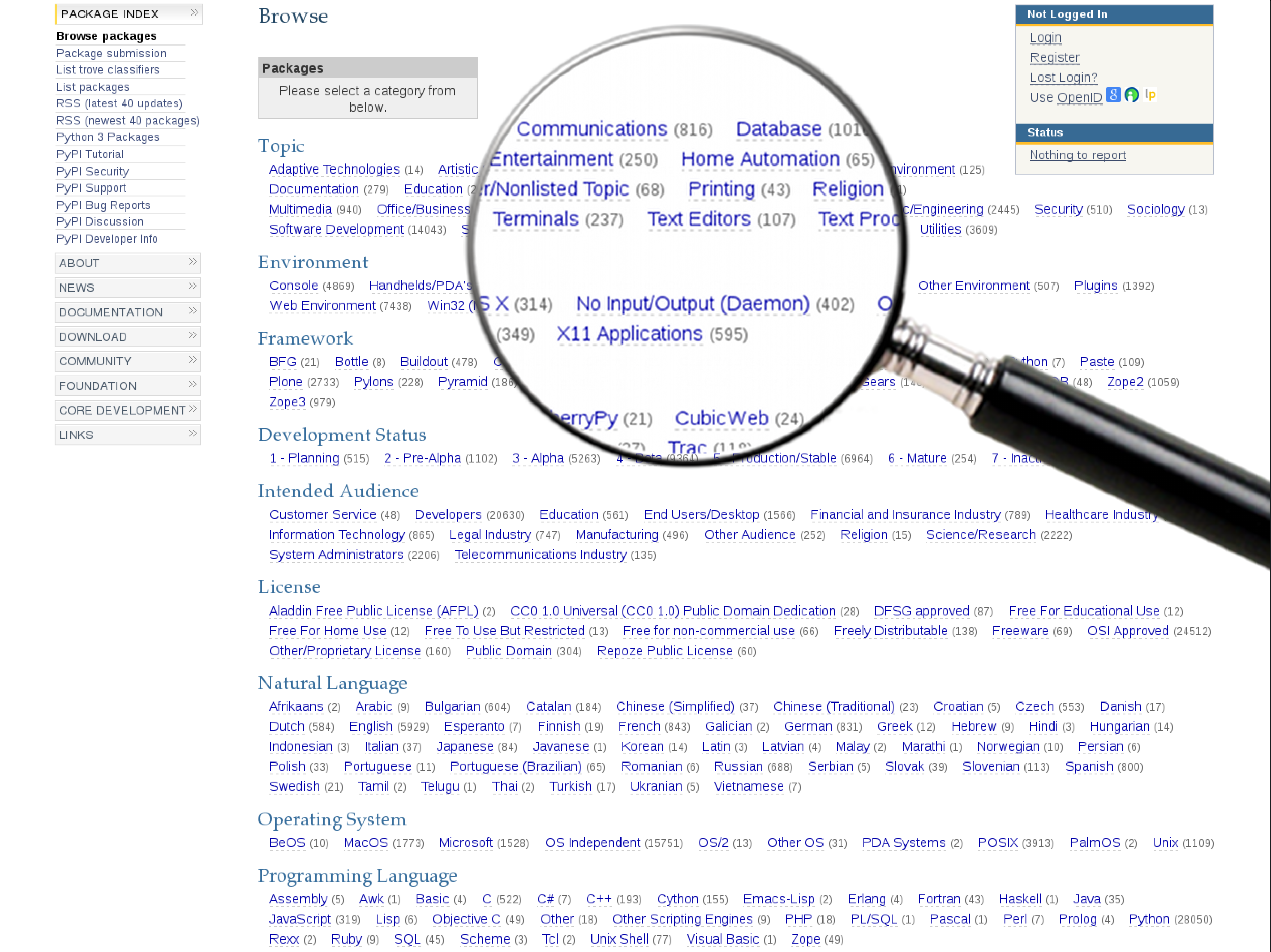
http://docs.python.org/2/library/

http://docs.python.org/2/py-modindex.html

modules

standing on the shoulders of giants

https://pypi.python.org/pypi?:action=browse



PACKAGE INDEX

Browse packages
Package submission
List trove classifiers
List packages
RSS (latest 40 updates)
RSS (newest 40 packages)
Python 3 Packages
PyPI Tutorial
PyPI Security
PyPI Support
PyPI Bug Reports
PyPI Discussion
PyPI Developer Info

ABOUT
NEWS
DOCUMENTATION
DOWNLOAD
COMMUNITY
FOUNDATION
CORE DEVELOPMENT
LINKS

Browse

Packages

Please select a category from below.

Topic

Adaptive Technologies (14) Artistic vironment (125)
Documentation (279) Education (2)
Multimedia (940) Office/Business c/Engineering (2445) Security (510) Sociology (13)
Software Development (14043) S Utilities (3609)

Communications (816) Database (101
Entertainment (250) Home Automation (65)
r/Nonlisted Topic (68) Printing (43) Religion (
Terminals (237) Text Editors (107) Text Proc

Environment

Console (4869) Handhelds/PDA's Other Environment (507) Plugins (1392)
Web Environment (7438) Win32 (

S X (314) No Input/Output (Daemon) (402) O
(349) X11 Applications (595)

Framework

BFG (21) Bottle (8) Buildout (478) C thon (7) Paste (109)
Plone (2733) Pylons (228) Pyramid (186) Gears (14 B (48) Zope2 (1059)
Zope3 (979)

erryPy (21) CubicWeb (24)
(27) Trac (119)

Development Status

1 - Planning (515) 2 - Pre-Alpha (1102) 3 - Alpha (5263) 4 - Beta (9364) 5 - Production/Stable (6964) 6 - Mature (254) 7 - Inact

Intended Audience

Customer Service (48) Developers (20630) Education (561) End Users/Desktop (1566) Financial and Insurance Industry (789) Healthcare Industry
Information Technology (865) Legal Industry (747) Manufacturing (496) Other Audience (252) Religion (15) Science/Research (2222)
System Administrators (2206) Telecommunications Industry (135)

License

Aladdin Free Public License (AFPL) (2) CC0 1.0 Universal (CC0 1.0) Public Domain Dedication (28) DFSG approved (87) Free For Educational Use (12)
Free For Home Use (12) Free To Use But Restricted (13) Free for non-commercial use (66) Freely Distributable (138) Freeware (69) OSI Approved (24512)
Other/Proprietary License (160) Public Domain (304) Repoze Public License (60)

Natural Language

Afrikaans (2) Arabic (9) Bulgarian (604) Catalan (184) Chinese (Simplified) (37) Chinese (Traditional) (23) Croatian (5) Czech (553) Danish (17)
Dutch (584) English (5929) Esperanto (7) Finnish (19) French (843) Galician (2) German (831) Greek (12) Hebrew (9) Hindi (3) Hungarian (14)
Indonesian (3) Italian (37) Japanese (84) Javanese (1) Korean (14) Latin (3) Latvian (4) Malay (2) Marathi (1) Norwegian (10) Persian (6)
Polish (33) Portuguese (11) Portuguese (Brazilian) (65) Romanian (6) Russian (688) Serbian (5) Slovak (39) Slovenian (113) Spanish (800)
Swedish (21) Tamil (2) Telugu (1) Thai (2) Turkish (17) Ukranian (5) Vietnamese (7)

Operating System

BeOS (10) MacOS (1773) Microsoft (1528) OS Independent (15751) OS/2 (13) Other OS (31) PDA Systems (2) POSIX (3913) PalmOS (2) Unix (1109)

Programming Language

Assembly (5) Awk (1) Basic (4) C (522) C# (7) C++ (193) Cython (155) Emacs-Lisp (2) Erlang (4) Fortran (43) Haskell (1) Java (35)
JavaScript (319) Lisp (6) Objective C (49) Other (18) Other Scripting Engines (9) PHP (18) PL/SQL (1) Pascal (1) Perl (7) Prolog (4) Python (28050)
Rexx (2) Ruby (9) SQL (45) Scheme (3) Tcl (2) Unix Shell (77) Visual Basic (1) Zope (49)

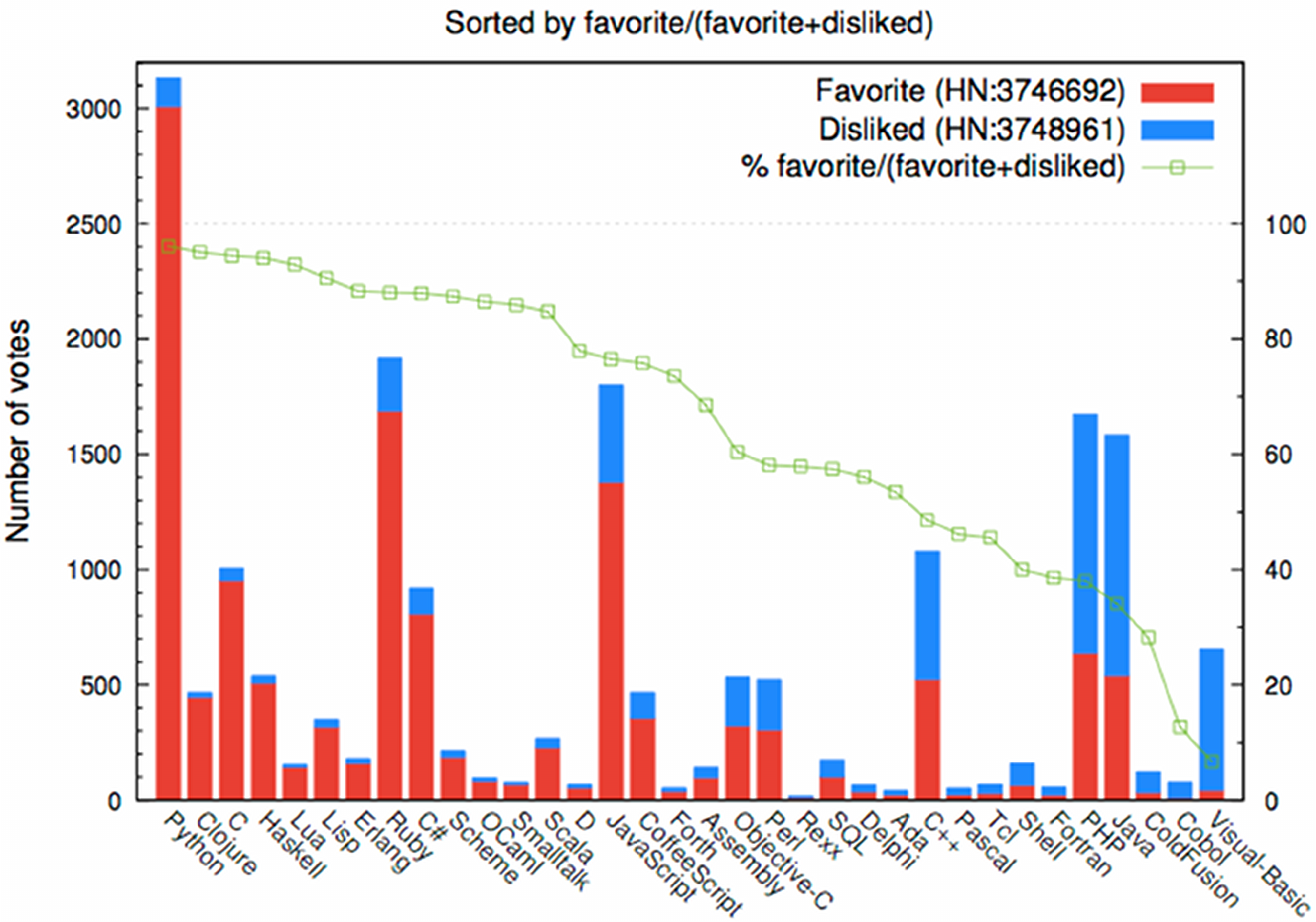
Not Logged In

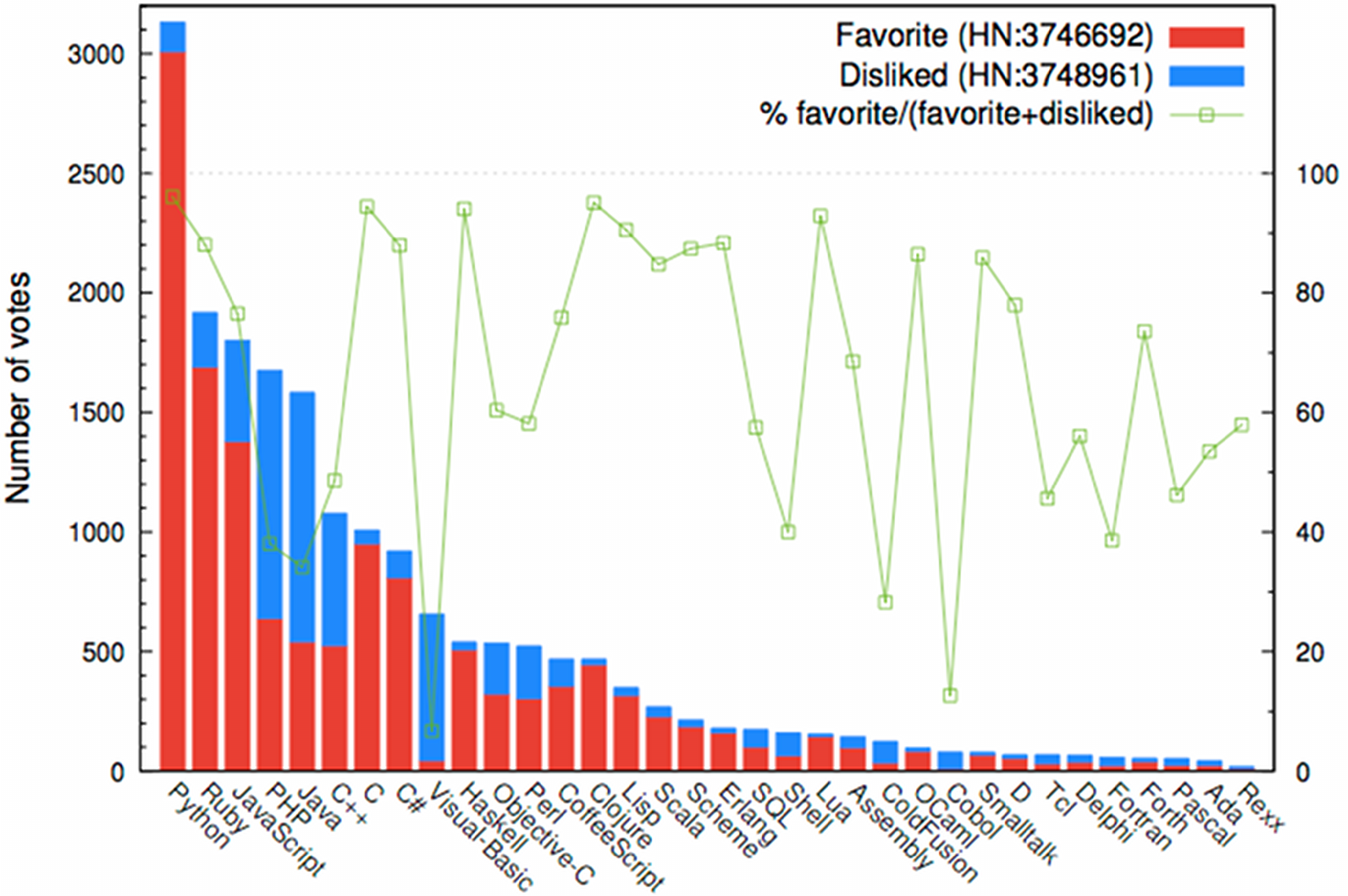
Login
Register
Lost Login?
Use OpenID

Status

Nothing to report

popularity


Sorted by favorite/(favorite+disliked)
Favorite (HN:3746692)
Disliked (HN:3748961)
% favorite/(favorite+disliked)
Number of votes
0
500
1000
1500
2000
2500
3000
0
20
40
60
80
100
Python
Clojure
C
Haskell
Lua
Lisp
Erlang
Ruby
C#
Scheme
OCaml
Smalltalk
Scala
D
JavaScript
CoffeeScript
Forth
Assembly
Objective-C
Perl
Rexx
SQL
Delphi
Ada
C++
Pascal
Tcl
Shell
Fortran
PHP
Java
ColdFusion
Cobol
Visual-Basic



HackerNews polls on favorite/disliked programming languages (Mon Mar 26 18:15:30 2012)

syntax

- elegant
- readable
- concise

```
print "Hello World!"
```

print the time

```
import datetime

print str(datetime.datetime.now())
```

Python 3

https://wiki.python.org/moin/Python2orPython3

variables

```
i = 5  # integer

f = 6.28  # float

s = "Hello World!"  # string

u = "Καλημέρα!"  # unicode

l = [0, 1, 1, 2, 3, 5, 8, 13, 21]  # list

d = {'first': 1, 'second': 2, 'third': 3}
# dictionary

b = True  # boolean
```

dynamic typing

no variable declaration

no variable initialization

dynamic typing

```
v = 5
print type(v)
<type 'int'>

v = v * 1.5
print type(v)
<type 'float'>

v = "Eve"
print type(v)
<type 'str'>
```

garbage collector

no memory management

OOP

Python is an object oriented programming language

string methods

```
s = "Hello World!"

print s.upper()
HELLO WORLD!

print s.lower()
hello world!

print s.find("o")
4

print s.replace("World", "everyone")
Hello everyone!
```

data structures

"Languages shape the way we think, or don't."

list methods

```
l = [0, 1, 1, 2, 3, 5, 8, 13, 21]

l.append(34)
print l
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

print l.pop()
34

print l.pop()
21

print l
[0, 1, 1, 2, 3, 5, 8, 13]
```

more list methods

```
l = [0, 1, 1, 2, 3, 5, 8, 13, 21]

l.reverse()
print l
[21, 13, 8, 5, 3, 2, 1, 1, 0]


l = [10, 5, 1, 3, 2, 4, 9, 8, 7, 6]

l.sort()
print l
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

list element referencing

```
l = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

print l[0], '+', l[1], '=', l[2]
1 + 2 = 3

print l[-1]
10

#         +---+---+---+---+
#         | l | i | s | t |
#         +---+---+---+---+
#         0   1   2   3   4
#        -4  -3  -2  -1
```

list slicing

```
l = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

print l[0:5]  # list[start:end]
[1, 2, 3, 4, 5]

print l[0:-5]
[1, 2, 3, 4, 5]

print l[1:-1]
[2, 3, 4, 5, 6, 7, 8,

print l[0:10:2]  # list[start:end:step]
[1, 3, 5, 7, 9]
```

dictionary methods

```
d = {'a': 1, 'b': 2, 'c': 3, 'd': 4}

print d['a']
1

d['e'] = 5
print d['e']
5

print d.keys()
['a', 'c', 'b', 'e', 'd']

print d.values()
[1, 3, 2, 5, 4]
```

if, elif, else

```
a, b = 1, 2

if a < b:
  print "a is less than b"
elif a > b:
  print "a is greater than b"
else:
  print "a and b are equal"
```

```
a is less than b
```

while

```
while True:
   print "Help! I'm stuck in a loop!"

temperature = 85
while temperature > 45:
   print temperature
   temperature -= 1
print "The tea is now cool enough."
```

for

```
for item in iterable_collection:
   # do something with item

for i in range(len(seq)):
   # do something with seq[i]

string = "Hello World!"
for x in string:
   print x

list_of_lists = [ [1, 2, 3], [4, 5, 6], [7, 8, 9]]
for list in list_of_lists:
   for x in list:
      print x
```

list comprehensions

```
squares = []
for x in range(10):
   squares.append(x**2)
print squares
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

# equivalent to

squares = [x**2 for x in range(10)]

# syntax
la = [expression for item in list if conditional]
```

functions

```
def get_max(a, b):
    """
    a and b are integers or floats
    """
    if a > b:
        return a
    else:
        return b


print get_max(1, 2)
2

print get_max(199, -54)
199
```

more functions

```
from __future__ import division
def get_average(i):
   """
   i is an iterable collection containing numbers
   """
   return sum(i) / len(i)


print get_average( [1, 2, 3, 4] )
2.5

print get_average( [0, 100] )
50.0
```

input

```
name = raw_input("What's your name? ")
print name

cost = input("How much is it? ")
pocket = input("How much do you have? ")
if cost > pocket:
  print "I'm sorry but you can't afford it."
else:
  print "Nice doing business with you."
```

output

```
print "\tHello everyone!\n"
    Hello everyone!

print  # print an empty line


s = "I am a string."
print s + " I really am.\n"  # concatenate strings
I am a string. I really am.

print "%s Again." % s
I am a string. Again.
```

file operations

```
echo "Creating a text file." > text_file.txt

f = open("text_file.txt", "r")  # read only
print f.read()
Creating a text file.

f = open("hello.txt", "w+")  # create if missing
f.write("Hello World! ")
f.close()
f = open("hello.txt", "a")  # append to the file
f.write("Hello again!\n")
f.close()

cat hello.txt
Hello World! Hello again!
```

using modules

```
import datetime

from __future__ import division

import numpy as np
```

Flask

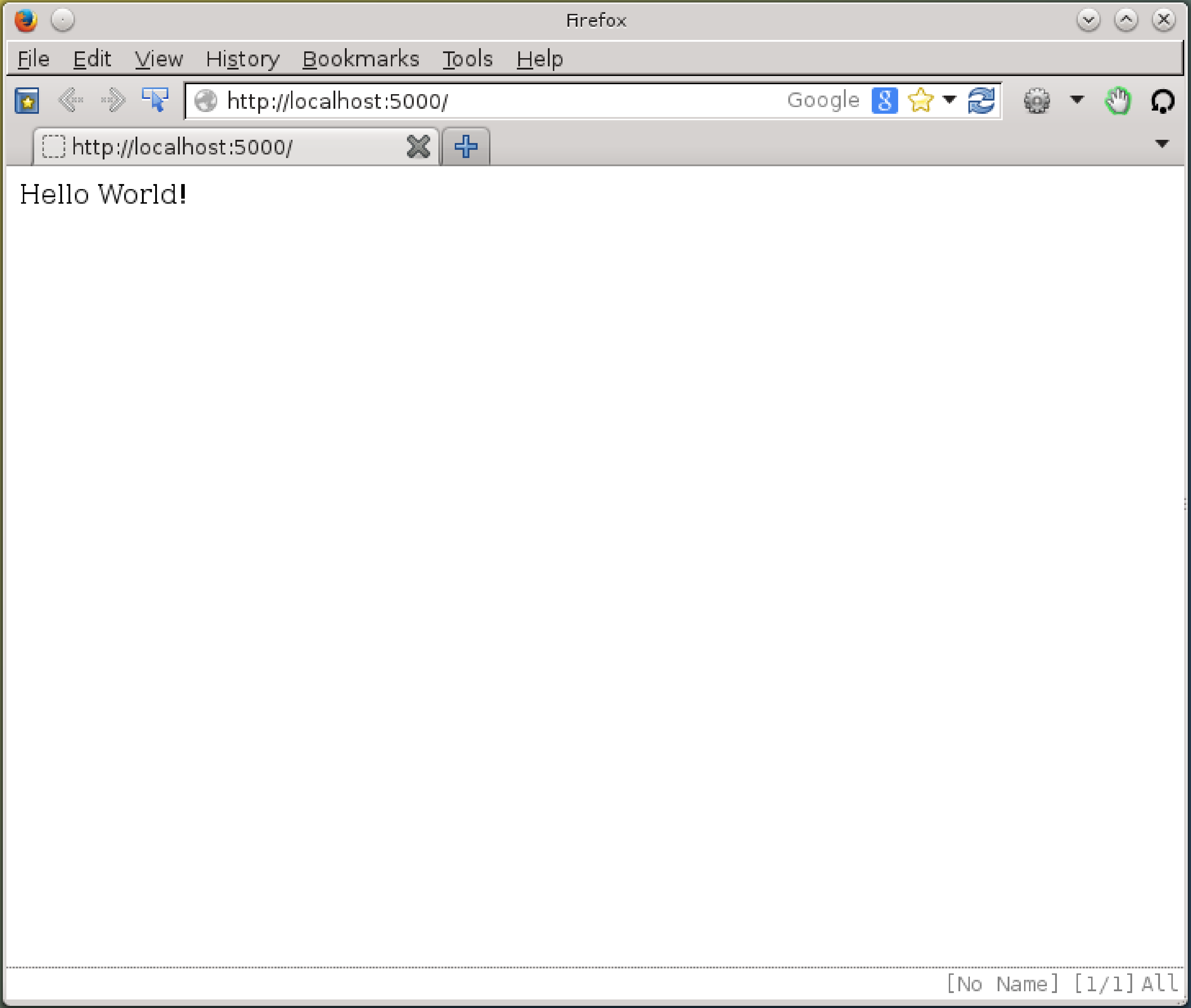
```
pip install Flask

from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
  return "Hello World!"

if __name__ == "__main__":
  app.run()

python server.py
 * Running on http://localhost:5000/
```


Firefox
File Edit View History Bookmarks Tools Help
http://localhost:5000/
Google
http://localhost:5000/

Hello World!

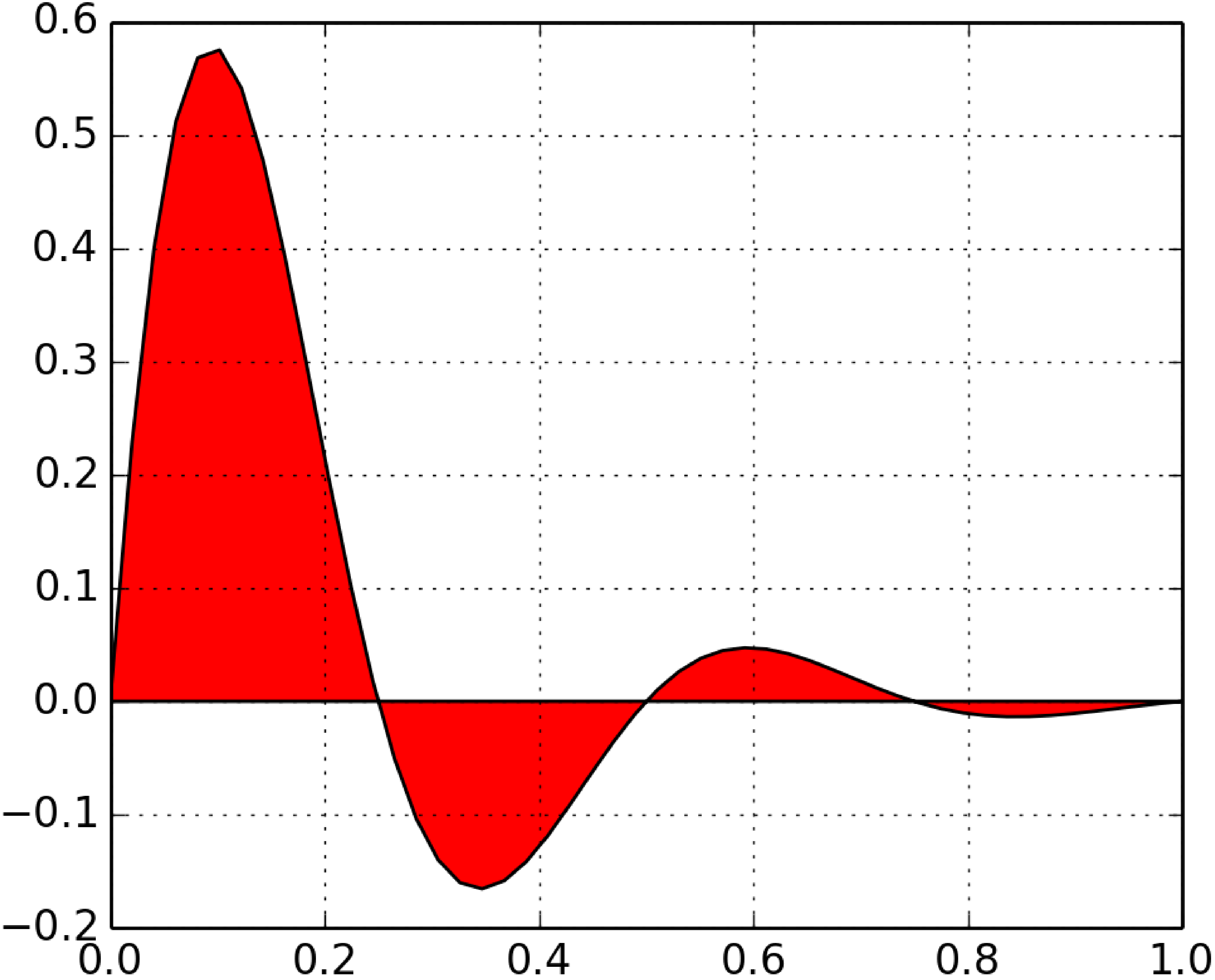
[No Name] [1/1] All

matplotlib

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 1)
y = np.sin(4 * np.pi * x) * np.exp(-5 * x)

plt.fill(x, y, 'r')
plt.grid(True)
plt.show()
```


0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
0.0
0.2
0.4
0.6
0.8
1.0

install

http://www.python.org/download/

editor

simple text editor

Vim for advanced usage

IDE

PyDev
http://pydev.org/

tools

IPython

resources

Python website
http://www.python.org/

documentation
http://docs.python.org/2/

Stack Overflow
http://stackoverflow.com/questions/tagged/python

Codecademy
http://www.codecademy.com/tracks/python

Think Python
http://www.greenteapress.com/thinkpython/

Dive Into Python
http://www.diveintopython.net/

Thank you!

@paraschas

Questions?

created for the hacker group of
the University of Thessaly, VolosHack

2014-03-08, Volos, Greece