

The Python Ecosystem

May 23, 2017

Overview



Python 2 vs. Python 3
Open Source Philosophy
Implementations
Work Time!

Python 2 vs. Python 3

Game Plan



The State of Python
Differences
Working with Both

The State of Python

Python 2.x is legacy
Python 3.x is the present and future of
the language

Timeline

Python 2 (released in 2000)

Python 2.7 is the last minor version of Python 2

Security and backport patches will continue until 2020

Python 3 (released in 2008)

7 stable releases and counting (3.0 through 3.6)

Actively developed

Why Change?

Intentional backwards-incompatibility allows larger fixes

"Clean up the Python language"

Guido van Rossum (BDFL) wanted something faster than
the usual deprecation process

Differences

Differences

Python 2.7 Code

Python 3.x Code

print

```
# print is a statement  
print "The answer is", 2*2
```

```
# Trailing comma suppresses  
newline  
print x,
```

```
# Prints a newline  
print
```

```
# Print to a file object  
print >>sys.stderr, "fatal  
error"
```

```
# Print a tuple  
print (x, y)
```

```
# print is a function  
print("The answer is", 2*2)
```

```
# Appends a space instead of a  
newline  
print(x, end=" ")
```

```
# You must call the function!  
print()
```

```
# Print to a file object  
print("fatal error",  
file=sys.stderr)
```

```
# Print a tuple  
print((x, y))
```

Views and Iterators instead of Lists

```
# .items/keys/values->list  
d = {'a': 1, 'b': 2}  
d.items()  
# => [('a', 1), ('b', 2)]  
d.iteritems() # => iterator
```

```
# map/filter -> list
```

```
# range->list, xrange->iterator  
range(4) # => [0, 1, 2, 3]  
xrange(4) # => iterator
```

```
zip([4, 1], 'py')  
# => [(4, 'p'), (1, 'y')]  
(list)
```

```
# .items/keys/values->view  
d = {'a': 1, 'b': 2}  
d.items() # => view
```

```
# map/filter -> iterator
```

```
# range->iterator  
range(4) # => iterator
```

```
zip([4, 1], 'py')  
# => iterator
```

input and raw_input

```
# input evaluates user input  
  
>>> x = input('Enter: ')  
Enter: 3 + 5  
  
>>> print x, type(x)  
8 <type 'int'>
```

```
# raw_input returns string  
  
>>> y = raw_input('Enter: ')  
Enter: 3 + 5  
  
>>> print y, type(y)  
3 + 5 <type 'str'>
```

```
# input returns string  
  
>>> x = input('Enter: ')  
Enter: 3 + 5  
  
>>> print x, type(x)  
3 + 5 <type 'str'>
```

And Many More

consolidation of integer types

new-style objects

integer division

native unicode support

exception syntax

standard library organization and naming

performance and memory

Working with Python 2 and Python 3

Translating Python 2 to Python 3

Use 2to3

"reads Python 2.x source code and applies a series of fixers
to transform it into valid Python 3.x code"

Works on almost all Python 2 code.

Writing Code for Python 2 and Python 3

What if you want to support a client running both Python 2 and Python 3?

Import from `__future__` ([docs](#))

`print_function`, `division`, `with_statement`, etc.

Use `six`

Useful abstractions, but fairly complicated

Probably only worth it for industry-scale projects

So...

Python 3 if you can
Python 2 if you must

Python in the Wild

Python at Stanford

CEE 245: Network Analysis for Urban Systems

COMM 382: Big Data and Causal Inference

CS 231N: Convolutional Neural Networks for Visual Recognition

EASTASN 105: Digital China: Computational Methods to Illuminate Society, Politics, and History

GENE 211: Genomics

LINGUIST 276: Quantitative Methods in Linguistics

MI 245: Computational Modeling of Microbial Communities

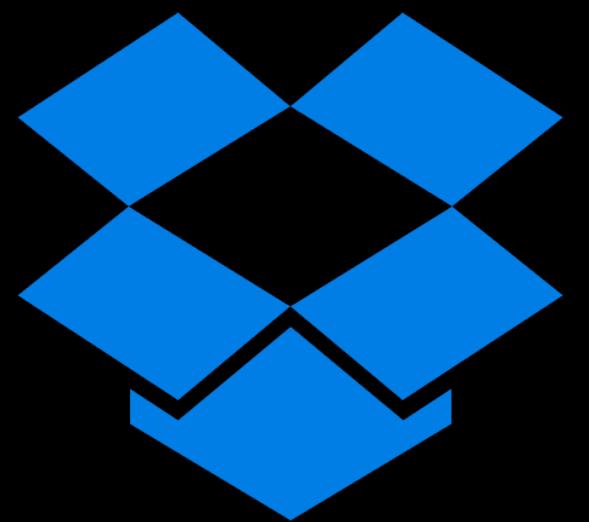
MS&E 448: Big Financial Data and Algorithmic Trading

PHYSICS 368: Computational Cosmology and Astrophysics

POLISCI 452: Text as Data

STATS 155: Statistical Methods in Computational Genetics

Python in Business



Dropbox



Quora

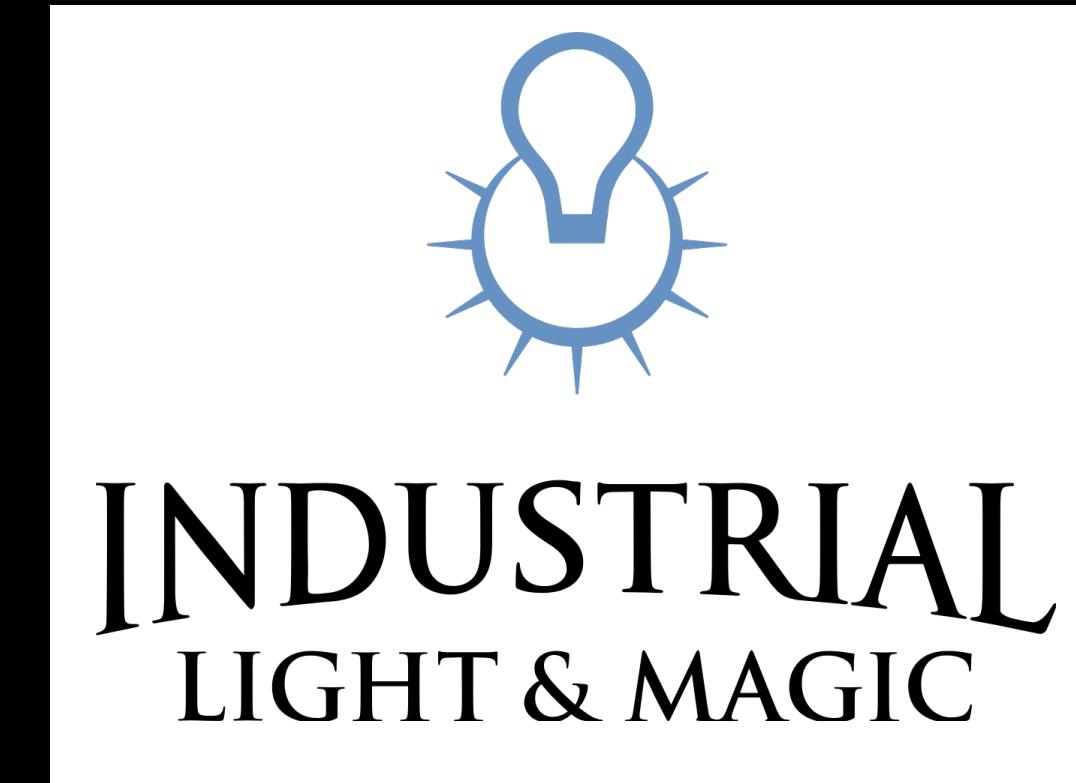
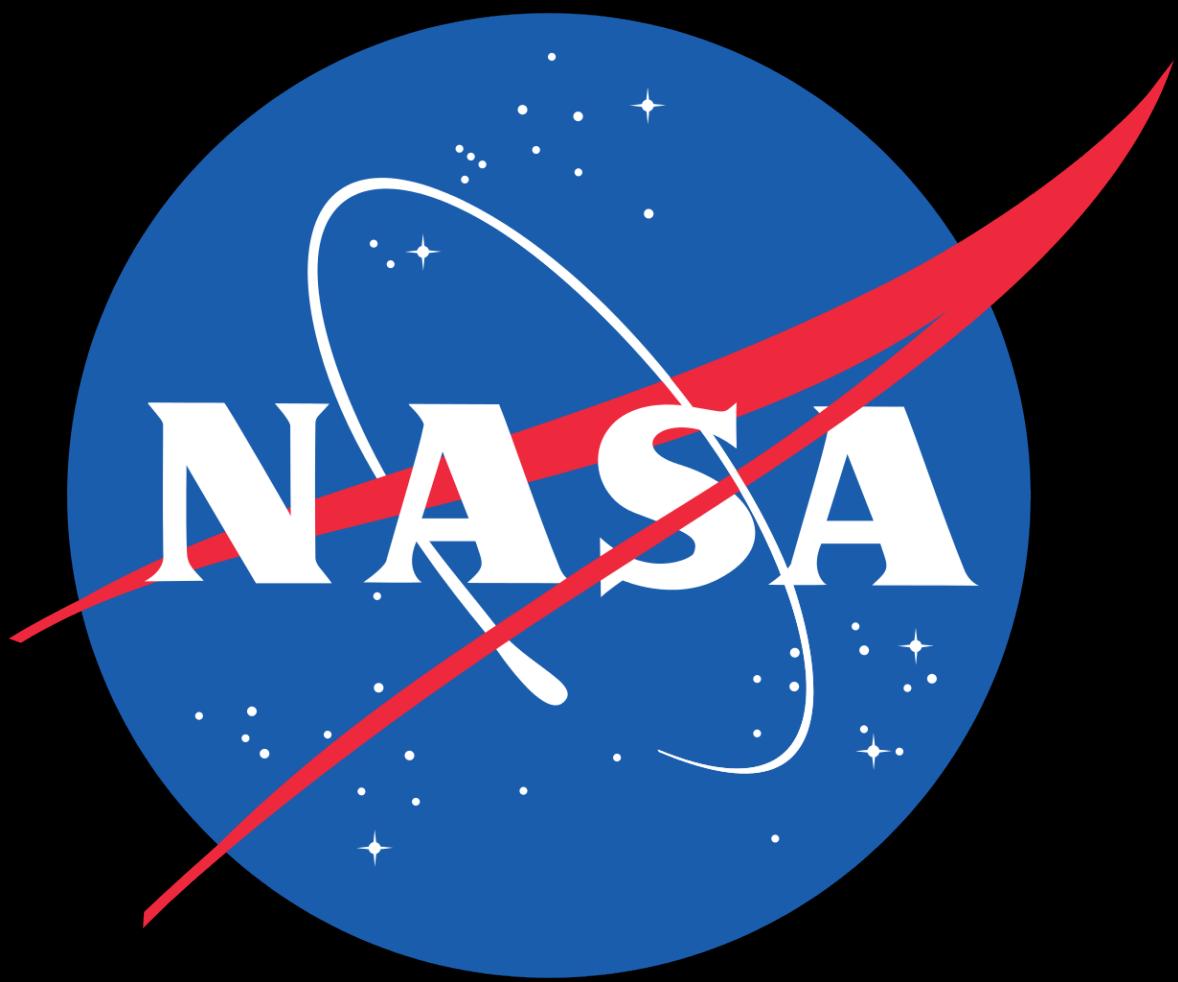


Google



Instagram

Other Python Users



Open Source Projects

Python encourages
open source software

kennethreitz / requests

Watch 1,002 Unstar 25,1

Code Issues 89 Pull requests 8 Projects 3 Wiki Insights

Branch: master requests / README.rst

kennethreitz Update README.rst

26 contributors

120 lines (85 sloc) 4.1 KB Raw Blame History

Requests: HTTP for Humans

pypi v2.14.2 license Apache 2.0 python 2.6, 2.7, 3.3, 3.4, 3.5, 3.6 build failing codecov 88% contributors 429 Say Thank You

Requests is the only Non-GMO HTTP library for Python, safe for human consumption.

tensorflow / tensorflow

Watch 5,167 Star 58,202 Fork 27,800

Code Issues 1,115 Pull requests 52 Projects 0 Insights

Branch: master tensorflow / README.md

av8ramit Updating version to 1.2.0-rc0

33 contributors

71 lines (57 sloc) 8.15 KB Raw Blame History

jakevdp / PythonDataScienceHandbook

Watch 370 Star 4,911 Fork 1,318

Code Issues 7 Pull requests 1 Projects 0 Wiki Insights

Python

Repositories related to the Python Programming language

https://www.python.org/

Repositories People 84

Pinned repositories

cpython

The Python programming language

Python 7.8k Stars 1.3k

mypy

Optional static typing for Python 2 and 3 (PEP484)

Python 2.5k Stars 312

pythondotorg

Source code for python.org

Python 699 Stars 239

peps

Python Enhancement Proposals

Python 465 Stars 170

typedsh

Collection of library stubs for Python, with static types

Python 417 Stars 249

devguide

The Python developer's guide

Python 228 Stars 65

Showing 575,843 available repository results

Create new file Upload files Find file Clone or download

Latest commit 9134c28 13 days ago

committed on GitHub TYPO: section 02.06 do->to ...

Fix typo on 02.06-Boolean-Arrays-and-Masks 13 days ago

Finalize table of contents & navigation links 6 months ago

add temporary files to gitignore 6 months ago

add Preface notebook 6 months ago

fix license :) 3 months ago

README: clarify requirements 4 months ago

MAINT: requirements numpy v1.11.1 6 months ago

python Data Science Handbook

Search repositories... Type: All Language: All

repository contains the entire Python Data Science Handbook, in the form of (free!) Jupyter notebooks.

Time-Out for
Announcements

Assignment 3

Stylize Recipe Book

Due *today* at midnight

If you have late days, might as well use them!

Advanced Topics

Next Week's Lecture

Advanced Topics Poll on Piazza

I'll talk about anything* you want!

* related to Python

Assignment 4 - Final Project

Due Friday Week 9 (June 2nd) at midnight

Top projects (decided by class-wide poll) invited to demo!

Back to Python!

Python Implementations

Python Implementations

"Python" refers to two separate concepts:

Language Specification

Language Implementation

Could we write our own Python interpreter?

Python Implementations - CPython

Reference Implementation of Python

Maintained by PSF

What we've used in this class

Core language features in C, most modules in Python

```
sredmond:stanfordpython$ python3
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 23 2015, 02:52:03)
[GCC X.Y.Z (.....) (.....)] on darwin
Type "help", "copyright", "credits" or "license" for more
information.
>>>
```

Python Implementations - PyPy

It's Python... in Python! (well, a restricted subset: RPython)

JIT (Just-In-Time) Compiler

C, CLI, JVM backends

Fast! >7x CPython on (certain) benchmark suites,

PyPy 5.17.1 is Python 2.7.13 compliant

PyPy3 5.7.1 is Python 3.5.3 compliant (in beta)

Python Implementations - Uncommon

Jython

Compiles Python to Java bytecode for JVM

Can import Java classes as modules

Jython2.7 fully compliant, Jython3.5 early development

IronPython/Python.NET

Integrates Python and .NET framework (for Windows)

Compliant with <= 2.7

Python Implementations - Uncommon

BeeWare

The IDEs of Python

Write a native iOS / Android app in Python

Python in Chrome/Safari/Firefox

Desktop Apps for macOS, Linux, Windows

MicroPython

Efficient Python implementation for microcontrollers

Next Time

Lab



Work day!

Start your final project

Next Week - Advanced Topics



Anything you want to hear
Vote in poll on Piazza

Work Time for Assignment 3!
(or Final Project)



Credit

Hitchhiker's Guide to Python

PSF PY2K vs. PY3K

PSF Alternate Implementations

