

# The Python Ecosystem

May 23, 2017

# Overview



# Overview

Python 2 vs. Python 3



# Overview



Python 2 vs. Python 3  
Open Source Philosophy

# Overview



Python 2 vs. Python 3  
Open Source Philosophy  
Implementations

# Overview



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

# Python 2 vs. Python 3

# Game Plan

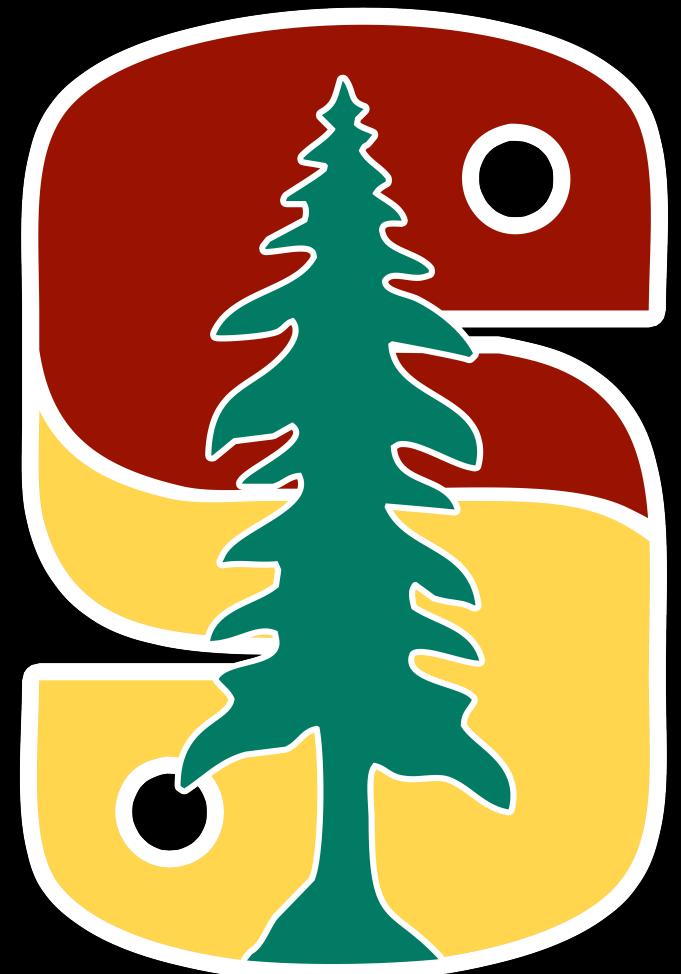


# Game Plan



The State of Python

# Game Plan



The State of Python  
Differences

# 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

# Timeline

**Python 2** (released in 2000)

# Timeline

**Python 2** (released in 2000)

Python 2.7 is the last minor version of Python 2

# 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

# 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)

# 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)

# 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?

# Why Change?

Intentional backwards-incompatibility allows larger fixes

# Why Change?

Intentional backwards-incompatibility allows larger fixes

"Clean up the Python language"

# 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

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

# print

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

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

# print

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

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

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

# print

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

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

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

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

# print

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

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

```
# Prints a newline  
print
```

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

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

# print

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

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

```
# Prints a newline  
print
```

```
# 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

```
# 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 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

```
# 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 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

```
# 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

```
# 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

# Views and Iterators instead of Lists

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

# Views and Iterators instead of Lists

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

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

# 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
```

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

# 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
```

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

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

# 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
```

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

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

# 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
```

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

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

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

# 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
```

# 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 and raw\_input

```
# input evaluates user input
```

# input and raw\_input

```
# input evaluates user input  
>>> x = input('Enter: ')  
Enter: 3 + 5
```

# input and raw\_input

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

# 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
```

# 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
```

# 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 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
```

# 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

And Many More  
consolidation of integer types

And Many More

consolidation of integer types

new-style objects

And Many More

consolidation of integer types

new-style objects

integer division

And Many More

consolidation of integer types

new-style objects

integer division

native unicode support

# And Many More

consolidation of integer types

new-style objects

integer division

native unicode support

exception syntax

# And Many More

consolidation of integer types

new-style objects

integer division

native unicode support

exception syntax

standard library organization and naming

# 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

# Translating Python 2 to Python 3

Use 2to3

# 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"

# 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

# Writing Code for Python 2 and Python 3

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

# 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](#))

# 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.

# 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](#)

# 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

# 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

# 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

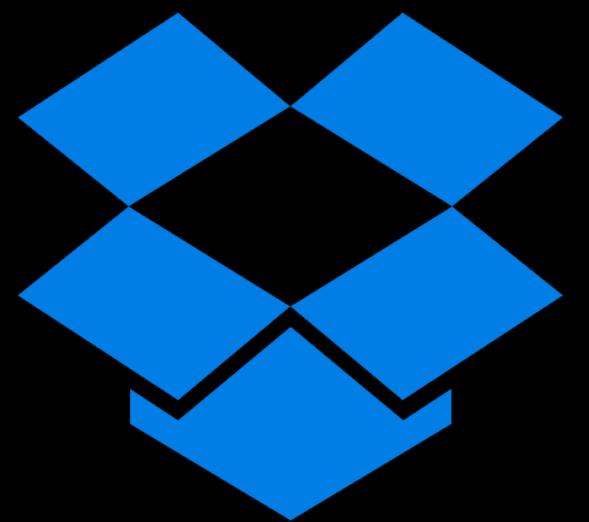
PHYSICS 368: Computational Cosmology and Astrophysics

POLISCI 452: Text as Data

STATS 155: Statistical Methods in Computational Genetics

# Python in Business

# Python in Business



Dropbox



Quora



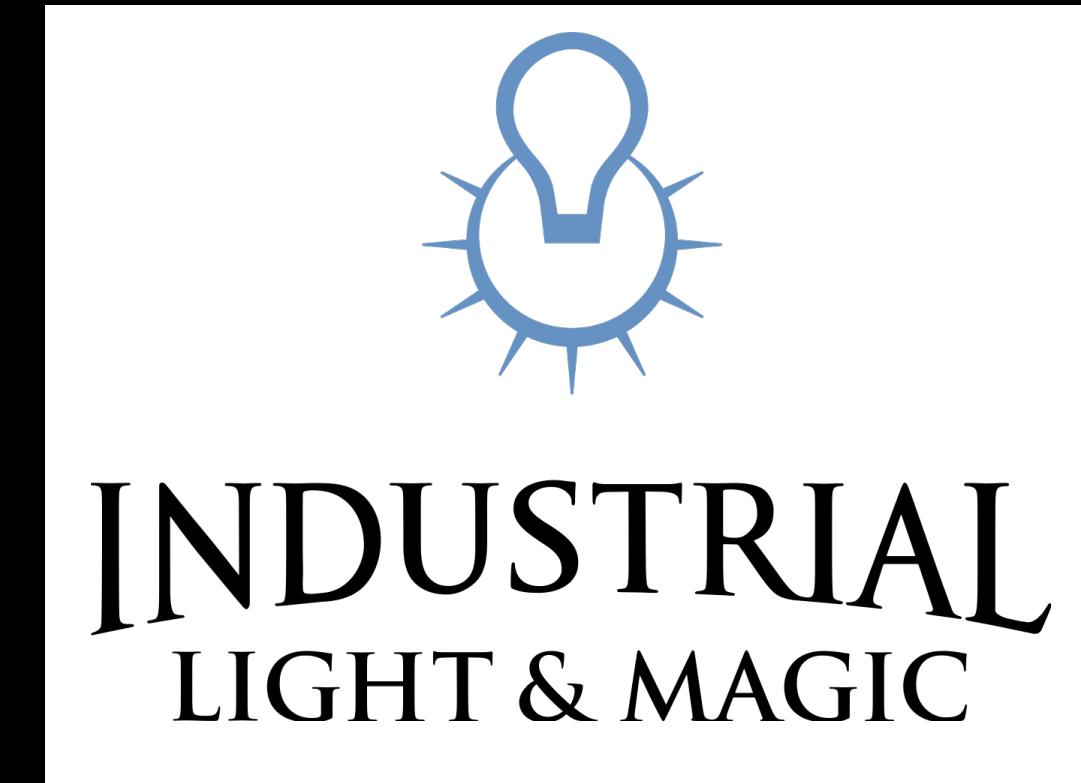
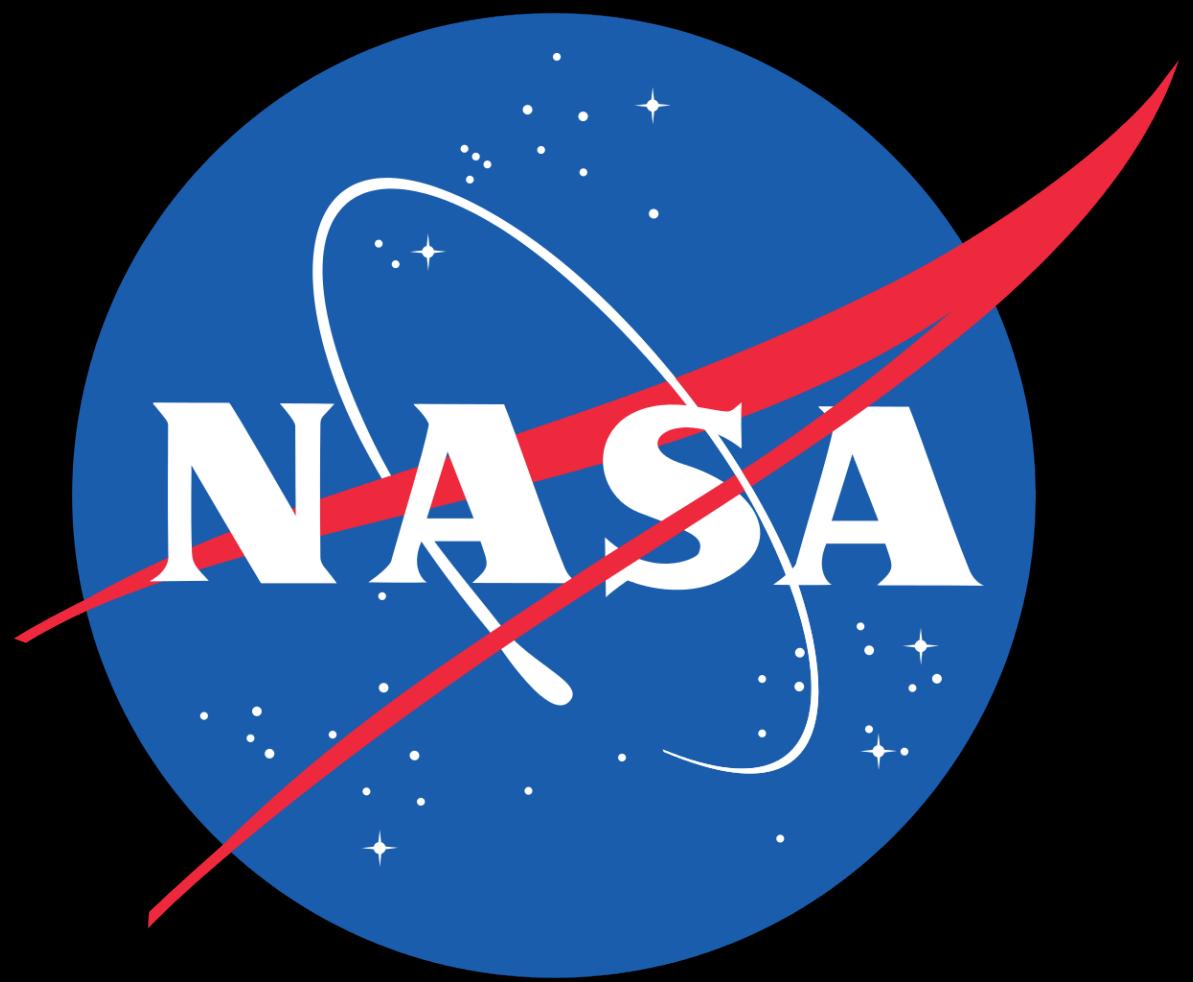
Google



Instagram

# Other Python Users

# Other Python Users



# Open Source Projects

Python encourages  
open source software



kennethreitz / requests

Watch 1,002 Unstar 25,117 Fork 4,561

Code Issues 89 Pull requests 8 Projects 3 Wiki Insights

Branch: master requests / README.rst Find file Copy path

 kennethreitz Update README.rst 379025a 18 days ago

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 Thanks !

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

**Warning:** Recreational use of the Python standard library for HTTP may result in dangerous side-effects, including: security vulnerabilities, verbose code, reinventing the wheel, constantly reading documentation, depression, headaches, or even death.

Behold, the power of Requests:

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
u'{"type": "User", ...}'
>>> r.json()
{u'disk_usage': 368627, u'private_gists': 484, ...}
```

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.

**Warning:** Recreational use of the Python standard library for HTTP may result in dangerous side-effects: security vulnerabilities, verbose code, reinventing the wheel, constantly reading documentation, depression, headaches, or even death.

Behold, the power of Requests:

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
u'{"type":"User"...'
>>> r.json()
{u'disk_usage': 368627, u'private_gists': 484, ...}
```

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



# TensorFlow

Linux CPU	Linux GPU	Mac OS CPU	Windows CPU	Android
build passing	build running	build running	build running	build passing

TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) that flow between them. This flexible architecture lets you deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device without rewriting code. TensorFlow also includes TensorBoard, a data visualization toolkit.

[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 Thanks!

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

**Warning:** Recreational use of the Python standard library for HTTP may result in dangerous side-effects: security vulnerabilities, verbose code, reinventing the wheel, constantly reading documentation, depression, headaches, or even death.

Behold, the power of Requests:

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
u'{"type": "User", ...}'
>>> r.json()
{u'disk_usage': 368627, u'private_gists': 484, ...}
```

[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

Jupyter Notebooks for the Python Data Science Handbook

196 commits 2 branches 0 releases 14 contributors MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

jakevdp committed on GitHub TYPO: section 02.06 do->to ... Latest commit 9134c28 13 days ago

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

tools Finalize table of contents & navigation links 6 months ago

.gitignore add temporary files to gitignore 6 months ago

LICENSE-CODE add Preface notebook 6 months ago

LICENSE-TEXT fix license :) 3 months ago

README.md README: clarify requirements 4 months ago

requirements.txt MAINT: requirements numpy v1.11.1 6 months ago

README.md

## Python Data Science Handbook

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

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	mypy	pythondotorg
The Python programming language	Optional static typing for Python 2 and 3 (PEP484)	Source code for python.org
Python 7.8k 1.3k	Python 2.5k 312	Python 699 239

peps	typedsh	devguide
Python Enhancement Proposals	Collection of library stubs for Python, with static types	The Python developer's guide
Python 465 170	Python 417 249	Python 228 65

Search repositories... Type: All Language: All

ebooks for the Python Data Science Handbook

commits 2 branches 0 releases 14 contributors MIT

New pull request Create new file Upload files Find file Clone or download

committed on GitHub TYPO: section 02.06 do->to ... Latest commit 9134c28 13 days ago

s 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

CODE add Preface notebook 6 months ago

TEXT fix license :) 3 months ago

md README: clarify requirements 4 months ago

ents.txt MAINT: requirements numpy v1.11.1 6 months ago

.md

## hon Data Science Handbook

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

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

# Assignment 3

## Stylize Recipe Book

# Assignment 3

**Stylize Recipe Book**

Due *today* at midnight

# Assignment 3

**Stylize Recipe Book**

Due *today* at midnight

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

# Advanced Topics

# Advanced Topics

Next Week's Lecture

# Advanced Topics

Next Week's Lecture

Advanced Topics Poll on Piazza

# Advanced Topics

## Next Week's Lecture

Advanced Topics Poll on Piazza

I'll talk about anything\* you want!

# 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

# Assignment 4 - Final Project

Due Friday Week 9 (June 2nd) at midnight

# 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 Implementations

"Python" refers to two separate concepts:

# Python Implementations

"Python" refers to two separate concepts:

Language Specification

# Python Implementations

"Python" refers to two separate concepts:

Language Specification

Language Implementation

# Python Implementations

"Python" refers to two separate concepts:

Language Specification

Language Implementation

Could we write our own Python interpreter?

# Python Implementations - CPython

# Python Implementations - CPython

## Reference Implementation of Python

# Python Implementations - CPython

Reference Implementation of Python

Maintained by PSF

# Python Implementations - CPython

Reference Implementation of Python

Maintained by PSF

What we've used in this class

# 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

# 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

# Python Implementations - PyPy

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

# Python Implementations - PyPy

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

JIT (Just-In-Time) Compiler

# Python Implementations - PyPy

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

JIT (Just-In-Time) Compiler

C, CLI, JVM backends

# 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,

# 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

# 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

# Python Implementations - Uncommon

Jython

# Python Implementations - Uncommon

## Jython

Compiles Python to Java bytecode for JVM

# Python Implementations - Uncommon

## Jython

Compiles Python to Java bytecode for JVM

Can import Java classes as modules

# 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

# 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

# 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)

# 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

# Python Implementations - Uncommon

BeeWare

# Python Implementations - Uncommon

## BeeWare

The IDEs of Python

# Python Implementations - Uncommon

## BeeWare

The IDEs of Python

Write a native iOS / Android app in Python

# Python Implementations - Uncommon

## BeeWare

The IDEs of Python

Write a native iOS / Android app in Python

Python in Chrome/Safari/Firefox

# 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

# 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

# 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



# Lab



Work day!

Start your final project

# Next Week - Advanced Topics

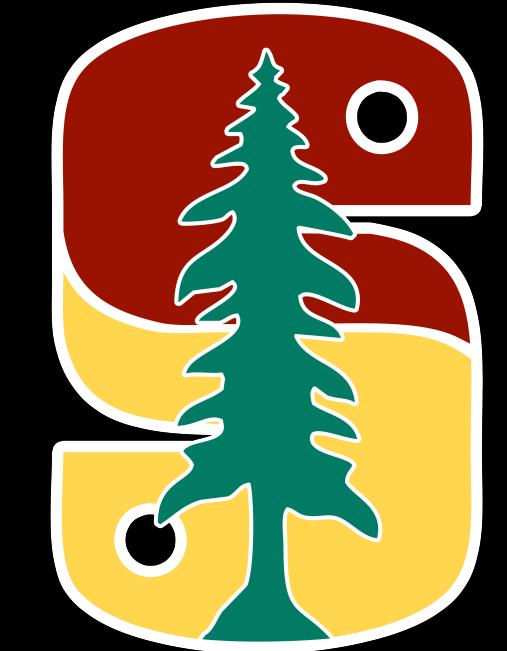


# Next Week - Advanced Topics



*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

