

Python Workshop

KnoxPy

September 5, 2019

Slides available at
github.com/knoxpy/presentations



Getting started

Download and install Python 3

The screenshot shows the Python.org homepage with a focus on the 'Downloads' section. The URL in the browser bar is www.python.org. The main navigation menu includes Python, PSF, Docs, PyPI, Jobs, and Community. Below the menu is the Python logo and a search bar with 'Search' and 'GO' buttons. A 'Donate' button is also visible. The 'Downloads' tab is selected, showing a sidebar with links for All releases, Source code, Windows, Mac OS X, Other Platforms, License, and Alternative Implementations. The main content area features a 'Download for Mac OS X' section with a prominent 'Python 3.7.4' button. Below it, text states: 'Not the OS you are looking for? Python can be used on many operating systems and environments.' and a link to 'View the full list of downloads.' At the bottom of the page, there are four cards: 'Get Started', 'Download', 'Docs', and 'Jobs'. The 'Download' card highlights Python 3.7.4 as the latest version. A note at the bottom encourages opening the download link in a new tab.

For loop on
>>> numbers =
>>> product =
>>> for number in numbers:
... product *= number
...
>>> print('The product is', product)
The product is 60

All releases

Source code

Windows

Mac OS X

Other Platforms

License

Alternative Implementations

Download for Mac OS X

Python 3.7.4

Not the OS you are looking for? Python can be used on many operating systems and environments.

[View the full list of downloads.](#)

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.

Download
Python source code and installers are available for download for all versions!

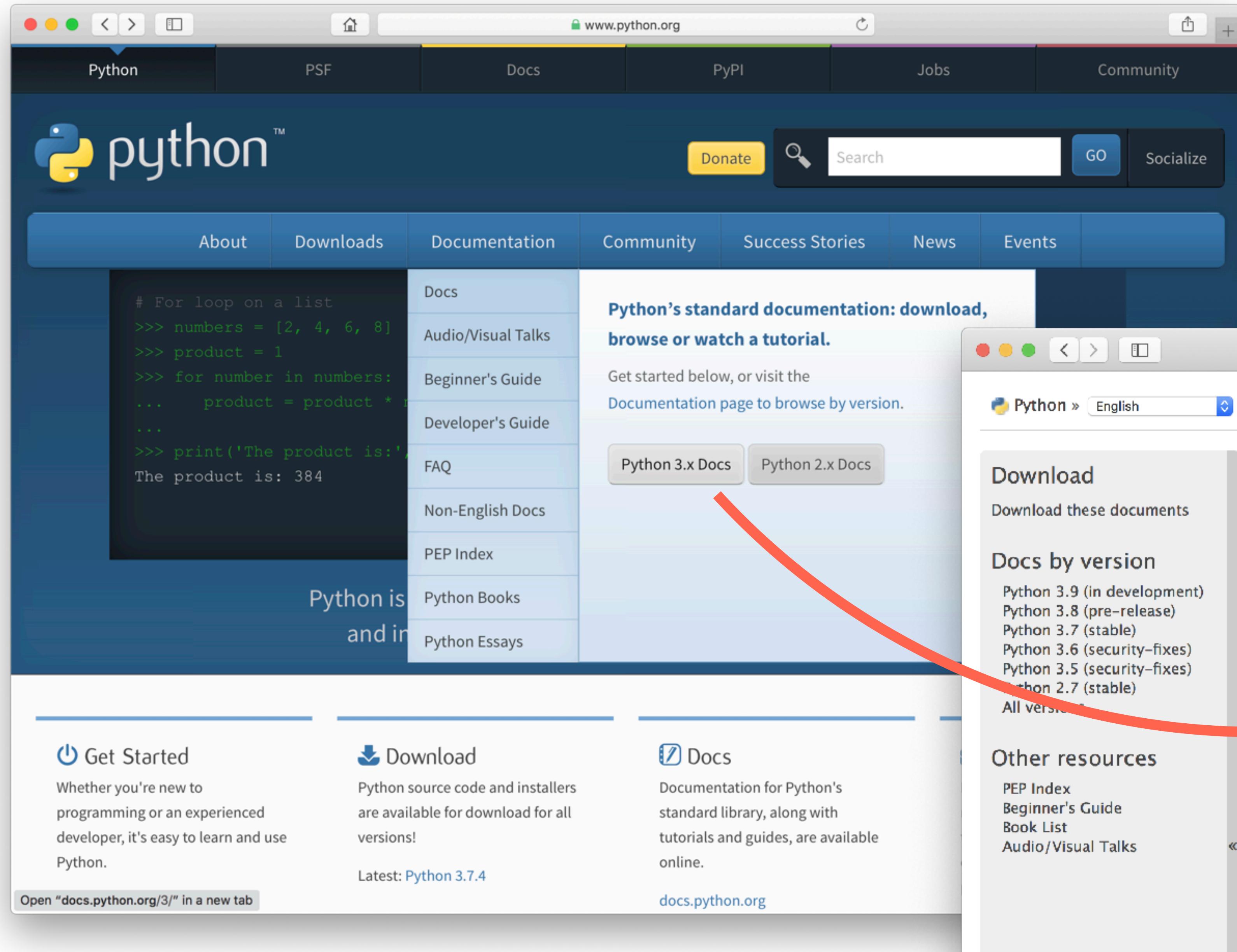
Latest: [Python 3.7.4](#)

[Open "https://www.python.org/ftp/python/3.7.4/python-3.7.4-macosx10.9.pkg" in a new tab](https://www.python.org/ftp/python/3.7.4/python-3.7.4-macosx10.9.pkg)

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 **relaunched community-run job board** is the place to go.

Python tutorial



The screenshot shows the Python.org homepage with the "Documentation" menu item highlighted in blue. A red arrow points from the "Documentation" link on the main menu down to the "Docs" section in the sidebar.

Main Navigation Bar:

- Python
- PSF
- Docs
- PyPI
- Jobs
- Community

Header:

- python™ logo
- Donate button
- Search bar
- GO button
- Socialize button

Top-level Links:

- About
- Downloads
- Documentation
- Community
- Success Stories
- News
- Events

Documentation Sidebar:

- # For loop on a list


```
>>> numbers = [2, 4, 6, 8]
>>> product = 1
>>> for number in numbers:
...     product = product * number
...
>>> print('The product is:', product)
The product is: 384
```
- Docs
- Audio/Visual Talks
- Beginner's Guide
- Developer's Guide
- FAQ
- Non-English Docs
- PEP Index
- Python Books
- Python Essays

Content Area:

Python's standard documentation: download, browse or watch a tutorial.

Get started below, or visit the [Documentation page](#) to browse by version.

[Python 3.x Docs](#) [Python 2.x Docs](#)

Footer:

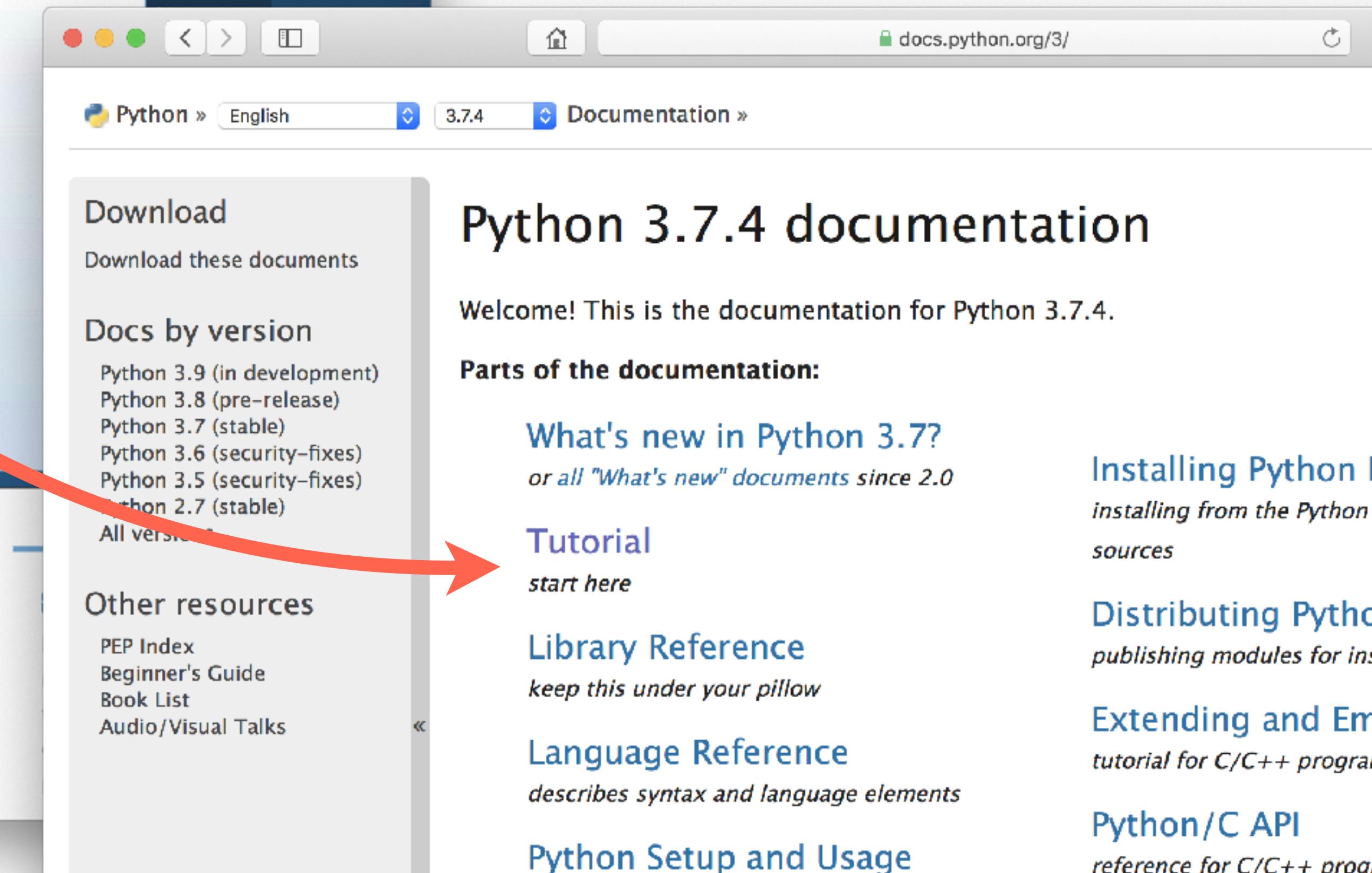
Python is and in

[Get Started](#)

[Download](#)

[Docs](#)

Open "docs.python.org/3/" in a new tab



The screenshot shows the Python 3.7.4 documentation page. A red arrow points from the "Python 3.x Docs" link on the Python.org homepage to the "Python 3.7.4 Documentation" link on this page.

Header:

- Python
- English
- 3.7.4
- Documentation

Content:

Python 3.7.4 documentation

Welcome! This is the documentation for Python 3.7.4.

Parts of the documentation:

- What's new in Python 3.7?** *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**

Download:

Download these documents

Docs by version:

- Python 3.9 (in development)
- Python 3.8 (pre-release)
- Python 3.7 (stable)
- Python 3.6 (security-fixes)
- Python 3.5 (security-fixes)
- Python 2.7 (stable)
- All versions

Other resources:

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

Installing Python
Installing from the Python sources

Distributing Python
publishing modules for installation

Extending and Embedding Python
tutorial for C/C++ programmers

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

Python for science

The screenshot shows the homepage of the Anaconda Distribution website (www.anaconda.com/distribution/). The page has a green header with the Anaconda logo and navigation links for Products, Why Anaconda?, Solutions, Resources, Company, Download, and Search. A large green banner features the text "Anaconda Distribution" and "The World's Most Popular Python/R Data Science Platform" with a "Download" button. Below the banner, a text block explains the benefits of the Anaconda Distribution, mentioning its use for Python/R data science and machine learning on various operating systems. To the right, there is a grid of logos for various data science libraries and tools, including Jupyter, Spyder, NumPy, SciPy, Numba, pandas, DASK, Bokeh, HoloViews, Datashader, Matplotlib, scikit-learn, TensorFlow, H2O.ai, and CONDA. At the bottom, icons indicate compatibility with Windows, macOS, and Linux.

Anaconda Distribution

The World's Most Popular Python/R Data Science Platform

Download

The open-source [Anaconda Distribution](#) is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download 1,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with [Conda](#)
- Develop and train machine learning and deep learning models with [scikit-learn](#), [TensorFlow](#), and [Theano](#)
- Analyze data with scalability and performance with [Dask](#), [NumPy](#), [pandas](#), and [Numba](#)
- Visualize results with [Matplotlib](#), [Bokeh](#), [Datashader](#), and [Holoviews](#)

Windows | macOS | Linux

Python programming language

Prompts, comments, and variables

Command prompt input is represented by **>>>**

Command prompt output is represented by absence of **>>>**

Comments in Python begin with the **#** character

The equal sign **=** is used to assign a value to a variable

```
>>> 2 + 2  
4  
>>> 50 - 5*6  
20
```

```
# this is the first comment  
spam = 1 # and this is the second comment  
# ... and now a third!  
text = "# This is not a comment because it's inside quotes."
```

```
>>> width = 20  
>>> height = 5 * 9  
>>> width * height  
900
```

Numbers

```
>>> 2 + 2          >>> 5 * 3 + 2    # result * divisor + remainder  
4                      17  
  
>>> 50 - 5*6        >>> 5 ** 2       # 5 squared  
20                     25  
  
>>> (50 - 5*6) / 4  >>> 2 ** 7       # 2 to the power of 7  
5.0                    128  
  
>>> 8 / 5           >>> 4 * 3.75 - 1  
# division always returns a floating point number  
1.6                    14.0  
  
>>> 10 / 3           # classic division returns a float  
3.333333335  
  
>>> 17 // 3           # floor division discards the fractional part  
5  
  
>>> 17 % 3            # the % operator returns the remainder of the division  
2
```

Strings

```
>>> 'spam eggs'      # single quotes
'spam eggs'

>>> 'doesn\'t'       # use \' to escape the single quote...
"doesn't"

>>> "doesn't"        # ...or use double quotes instead
"doesn't"

>>> """Yes," they said.' 
"""Yes," they said.

>>> """\\"Yes,\\" they said."
"""Yes," they said.

>>> """Isn\'t," they said.'
"""Isn\'t," they said.

>>> 3 * 'un' + 'ium'   # 3 times 'un', followed by 'ium'
'unununium'

>>> 'Py' 'thon'        # two or more string literals automatically combine
'Python'

>>> prefix + 'thon'    # concatenate variables or a variable and a literal
'Python'
```

Lists

```
>>> squares = [1, 4, 9, 16, 25]
>>> squares
[1, 4, 9, 16, 25]

>>> squares[0] # indexing returns the item
1

>>> squares[-1]
25

>>> squares[-3:] # slicing returns a new list
[9, 16, 25]

>>> squares + [36, 49, 64, 81, 100]
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

>>> cubes = [1, 8, 27, 65, 125]
>>> cubes[3] = 64
>>> cubes
[1, 8, 27, 64, 125]

>>> letters = ['a', 'b', 'c', 'd']
>>> len(letters)
4
```

Tuples

```
>>> t = 12345, 54321, 'hello!'

>>> t[0]
12345

>>> t
(12345, 54321, 'hello!')

>>> u = t, (1, 2, 3, 4, 5) # tuples may be nested
>>> u
((12345, 54321, 'hello!'), (1, 2, 3, 4, 5))

>>> t[0] = 88888           # tuples are immutable
TypeError: 'tuple' object does not support item assignment

>>> empty = ()            # construct an empty tuple
>>> len(empty)
0

>>> singleton = 'hello',   # construct a one item tuple
>>> len(singleton)
1
>>> singleton
('hello',)

>>> t = 12345, 54321, 'hello!' # tuple packing
>>> x, y, z = t                # tuple unpacking
```

Dictionaries

```
>>> tel = {'jack': 4098, 'sape': 4139}
>>> tel['guido'] = 4127
>>> tel
{'jack': 4098, 'sape': 4139, 'guido': 4127}

>>> tel['jack']
4098

>>> del tel['sape']
>>> tel['irv'] = 4127
>>> tel
{'jack': 4098, 'guido': 4127, 'irv': 4127}

>>> list(tel)
['jack', 'guido', 'irv']

>>> sorted(tel)
['guido', 'irv', 'jack']

>>> 'guido' in tel
True

>>> 'jack' not in tel
False
```

Sets

```
>>> basket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}
>>> basket
{'orange', 'banana', 'pear', 'apple'} # duplicates have been removed

>>> 'orange' in basket # fast membership testing
True
>>> 'crabgrass' in basket
False

>>> a = set('abracadabra')
>>> b = set('alacazam')
>>> a # unique letters in a
{'a', 'r', 'b', 'c', 'd'}

>>> a - b # letters in a but not in b
{'r', 'd', 'b'}

>>> a | b # letters in a or b or both
{'a', 'c', 'r', 'd', 'b', 'm', 'z', 'l'}

>>> a & b # letters in both a and b
{'a', 'c'}

>>> a ^ b # letters in a or b but not both
{'r', 'd', 'b', 'm', 'z', 'l'}
```

Control flow with “if” and “for” statements

```
>>> x = 42

>>> if x < 0:
...     x = 0
...     print('Negative changed to zero')
... elif x == 0:
...     print('Zero')
... elif x == 1:
...     print('Single')
... else:
...     print('More')
```

More

```
>>> words = ['cat', 'window', 'defenestrate']

>>> for w in words:
...     print(w, len(w))

cat 3
window 6
defenestrate 12

>>> for w in words[:]: # Loop over a slice copy of the entire list.
...     if len(w) > 6:
...         words.insert(0, w)

>>> words
['defenestrate', 'cat', 'window', 'defenestrate']
```

```
>>> a = ['Mary', 'had', 'a', 'little', 'lamb']
>>> for i in range(len(a)):
...     print(i, a[i])
```

```
0 Mary
1 had
2 a
3 little
4 lamb
```

Functions

```
def fib2(n):
"""
Return a list containing the Fibonacci
series up to n.
"""
result = []
a, b = 0, 1
while a < n:
    result.append(a)
    a, b = b, a+b
return result

>>> f100 = fib2(100)      # call it

>>> f100                  # write the result
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
```

```
def ask_ok(prompt, retries=4, reminder='Please try again!'):
    while True:
        ok = input(prompt)
        if ok in ('y', 'ye', 'yes'):
            return True
        if ok in ('n', 'no', 'nop', 'nope'):
            return False
        retries = retries - 1
        if retries < 0:
            raise ValueError('invalid user response')
        print(reminder)

>>> ask_ok('Do you really want to quit?')

>>> ask_ok('OK to overwrite the file?', 2)
>>> ask_ok('OK to overwrite the file?', 2, 'Come on, only yes or no!')
```

Classes

```
class Dog:

    def __init__(self, name):
        self.name = name
        self.tricks = []      # creates a new empty list for each dog

    def add_trick(self, trick):
        self.tricks.append(trick)

>>> d = Dog('Fido')

>>> e = Dog('Buddy')

>>> d.add_trick('roll over')

>>> e.add_trick('play dead')

>>> d.tricks
['roll over']

>>> e.tricks
['play dead']
```

Modules

17

fibo.py

```
# Fibonacci numbers module
# This code is saved into a file named fibo.py

def fib(n):      # write Fibonacci series up to n
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a+b
    print()

def fib2(n):     # return Fibonacci series up to n
    result = []
    a, b = 0, 1
    while a < n:
        result.append(a)
        a, b = b, a+b
    return result
```

Python interpreter

```
>>> import fibo
>>> fibo.fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
>>> fibo.fib2(100)
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
```

Python Package Index (PyPI)

18

The screenshot shows the homepage of the Python Package Index (PyPI) at pypi.org. The page has a blue header with the PyPI logo, a search bar, and navigation links for Help, Donate, Log in, and Register. The main content area features a large white banner with the text "Find, install and publish Python packages with the Python Package Index". Below the banner is a search bar with the placeholder "Search projects" and a magnifying glass icon. A link "Or [browse projects](#)" is also present. At the bottom of the main section, there are statistics: 194,619 projects, 1,450,470 releases, 2,129,503 files, and 366,564 users. The footer contains the Python Package Index logo and a brief description of what PyPI is and how it helps the community.

194,619 projects 1,450,470 releases 2,129,503 files 366,564 users

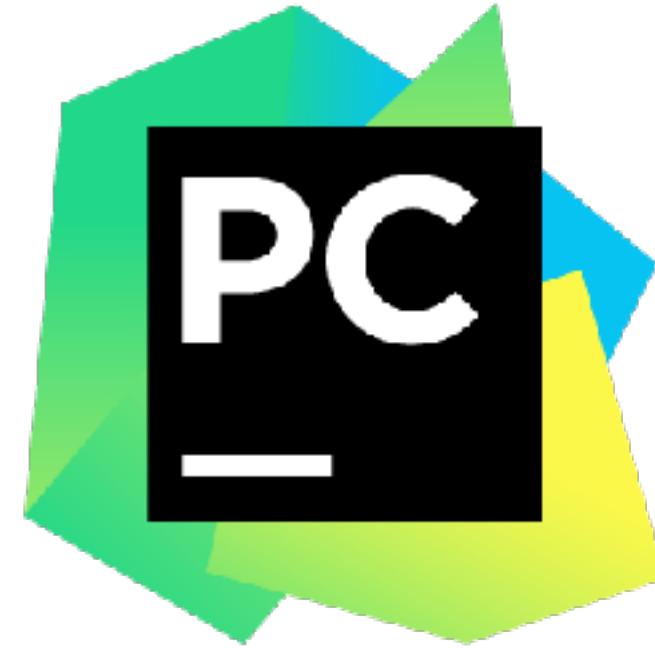
python™
Package
Index

The Python Package Index (PyPI) is a repository of software for the Python programming language. PyPI helps you find and install software developed and shared by the Python community. [Learn about installing packages ↗](#). Package authors use PyPI to distribute their software. [Learn how to package your Python code for PyPI ↗](#).

Resources

IDEs and Text Editors

20



PyCharm

<https://www.jetbrains.com/pycharm/>



Sublime Text

<https://www.sublimetext.com>



Spyder IDE

<https://github.com/spyder-ide/spyder>

IP[y]:
IPython

iPython

<https://ipython.org>



Jupyter Notebook

<http://jupyter.org>



Visual Studio Code

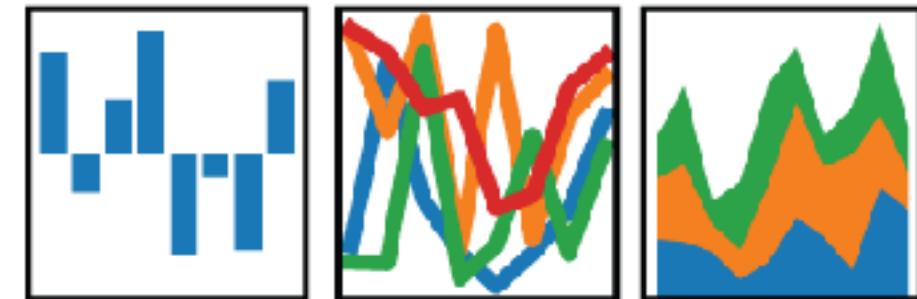
<https://code.visualstudio.com>

Modules (Packages)

21

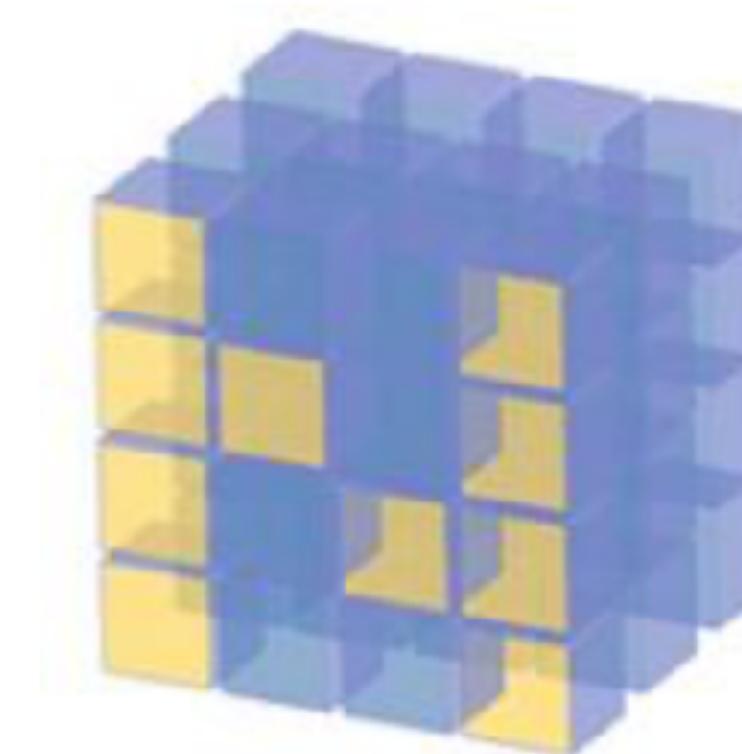
pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Pandas

<http://pandas.pydata.org>



NumPy

<http://www.numpy.org>



SciPy

<https://www.scipy.org>



Flask

<http://flask.pocoo.org>

matplotlib

Matplotlib

<http://matplotlib.org>



Requests

<http://docs.python-requests.org>

Modules (Packages)

22



MicroPython
<https://micropython.org>



CircuitPython
<https://github.com/adafruit/circuitpython>



SQLAlchemy
<https://www.sqlalchemy.org>



Bokeh
<https://bokeh.pydata.org>

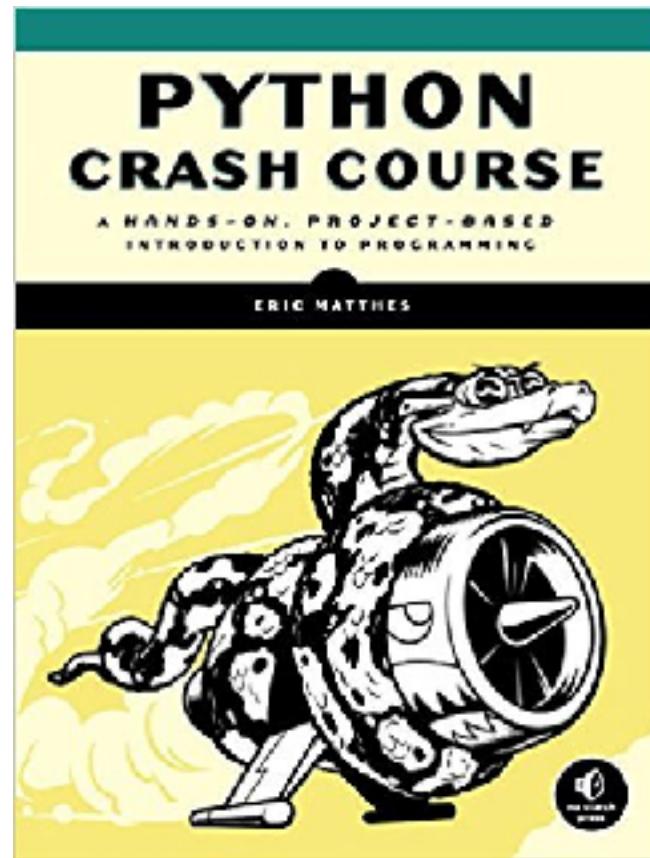


Django
<https://www.djangoproject.com>

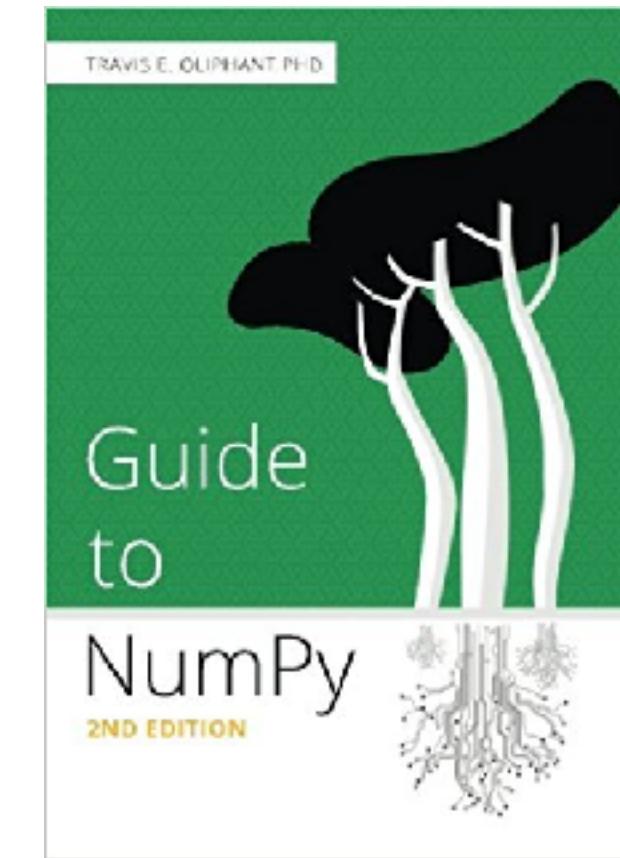


Sympy
<http://www.sympy.org>

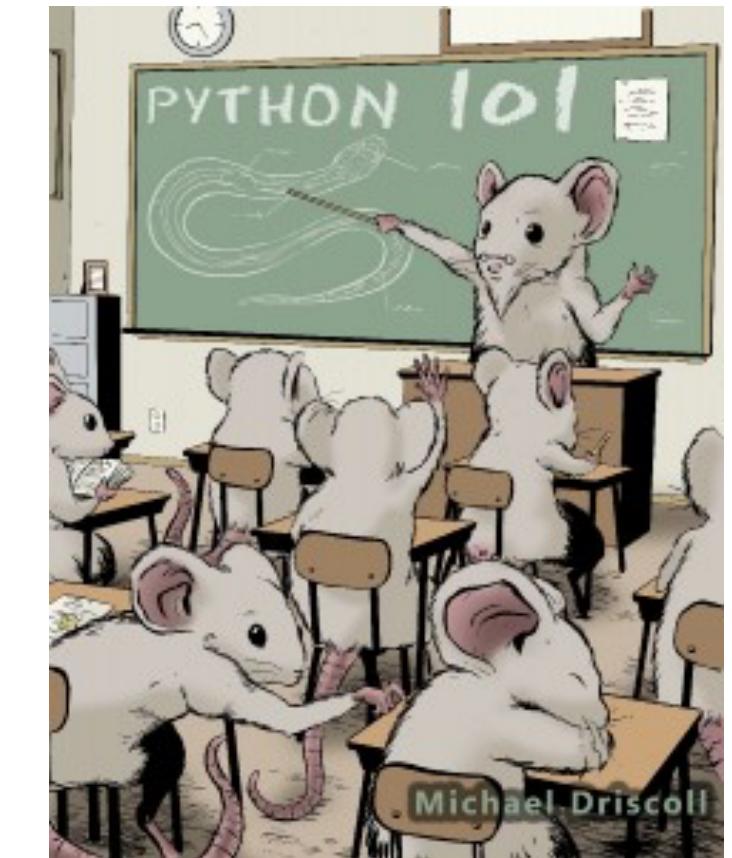
Books



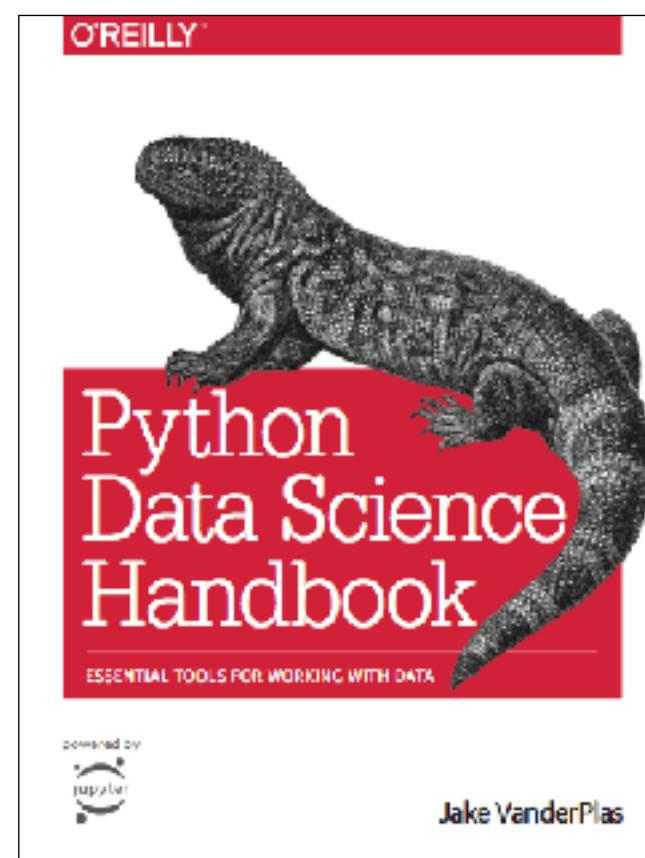
Python Crash Course
by Eric Matthes



Guide to NumPy
by Travis Oiphant

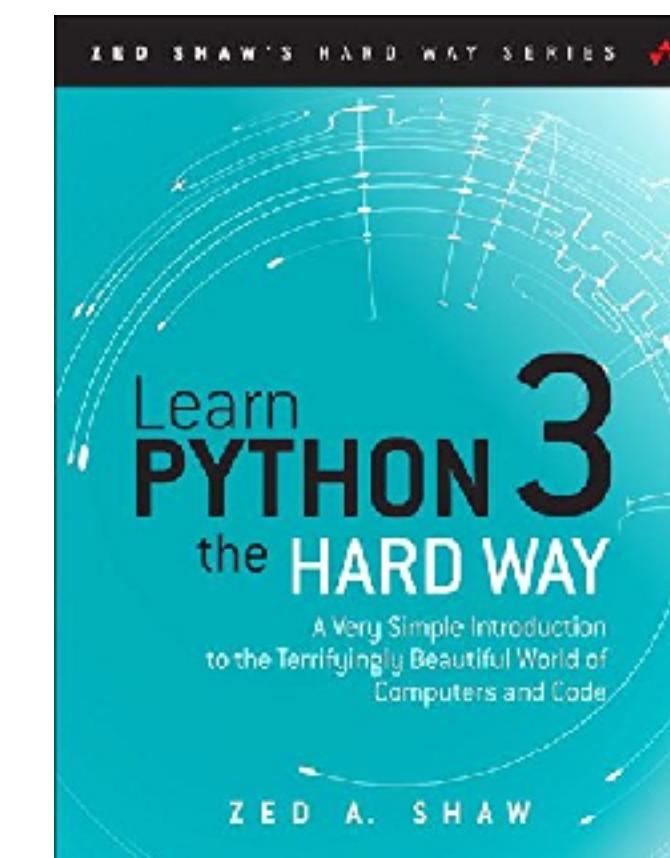


Python 101
by Michael Driscoll

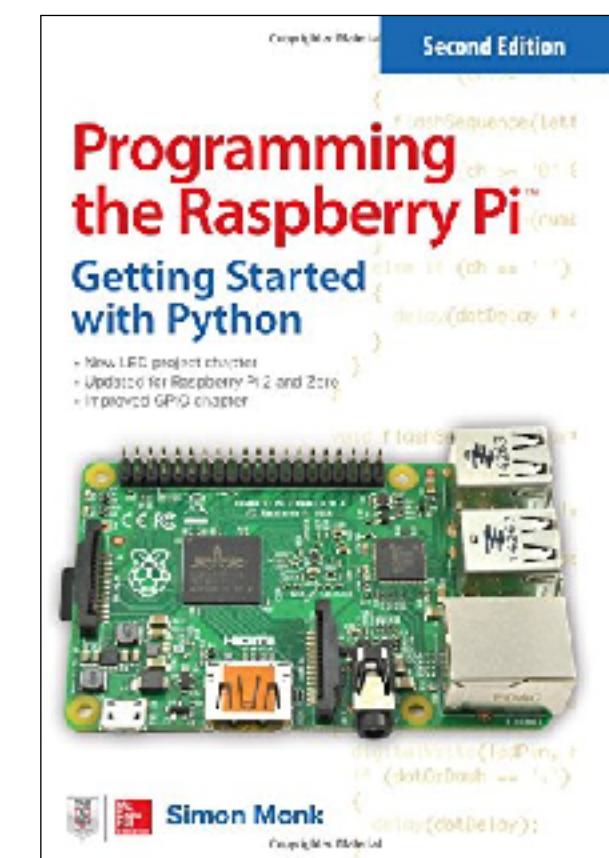


Python Data Science Handbook
by Jake VanderPlas

<https://jakevdp.github.io/PythonDataScienceHandbook/>



Learn Python 3 the Hard Way
by Zed Shaw



Programming the Raspberry Pi
by Simon Monk

Online Courses and Tutorials

24



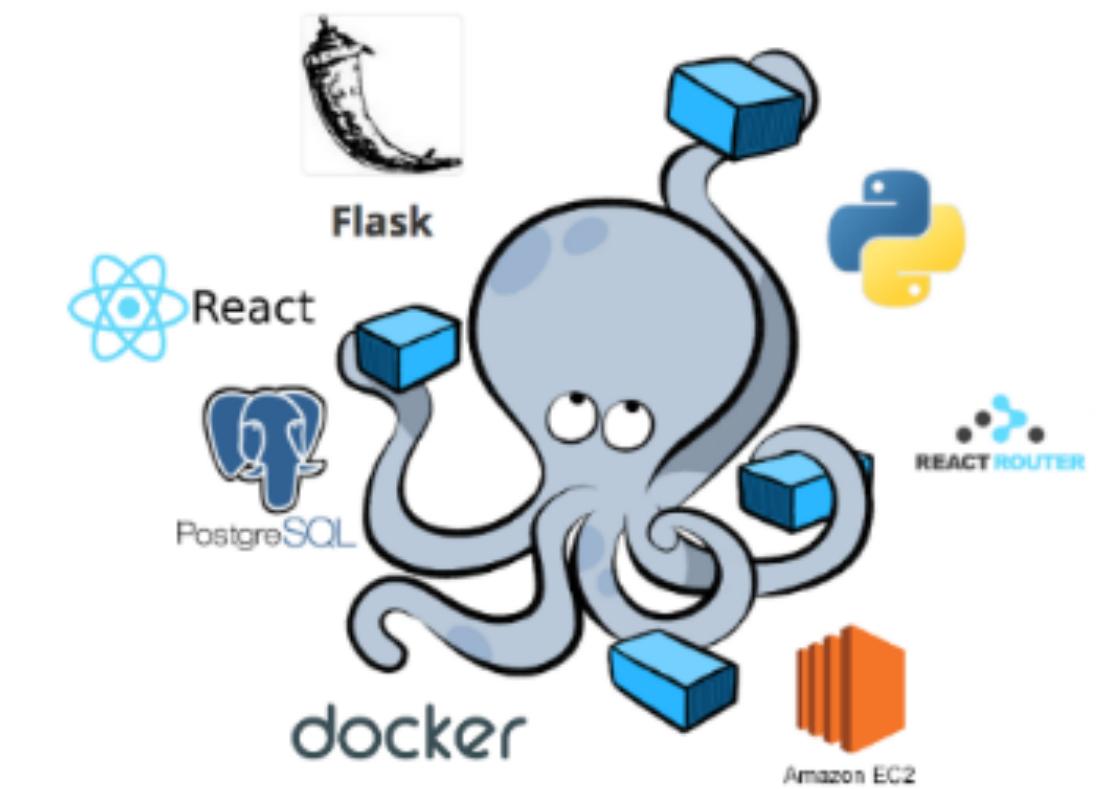
DataCamp

<https://www.datacamp.com>



Snakify

<https://snakify.org>



Test Driven Development

<http://testdriven.io>



LearnPython

<https://www.learnpython.org>



Coursera

<https://www.coursera.org>



Udacity

<https://www.udacity.com>

Conferences



SciPy2017

SciPy

scientific computing conference

<https://conference.scipy.org>



PyCon

largest gathering for open-source python

<https://us.pycon.org>



PyTennessee

regional conference in Nashville

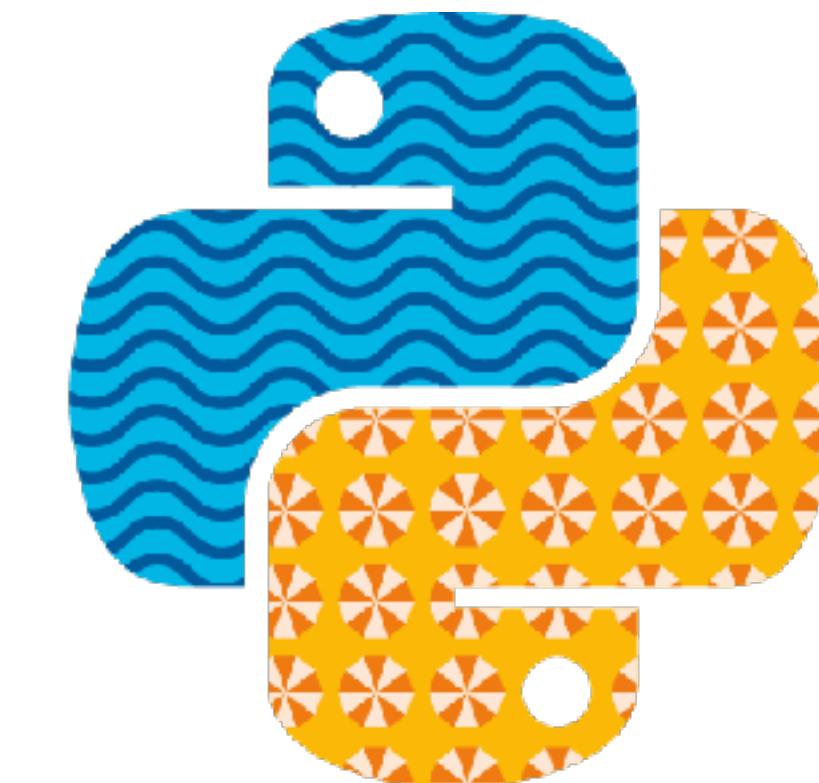
<https://www.pytennessee.org>



AnacondaCon

open data science conference

<https://anacondacon18.io>



EuroPython

largest European python conference

<https://ep2017.europython.eu/en/>



PyOhio

free annual python conference

<https://pyohio.org>

More Resources

26

The screenshot shows two views of the KnoxPy.org website. The top view is a static image of the homepage, while the bottom view is a live browser window showing the 'Resources' page. A red arrow points from the text 'Scroll down' to the bottom browser window, indicating where to look for more resources.

KnoxPy 🐍💻
Python users group in Knoxville, TN

Welcome Pythonistas! The KnoxPy group meets the first month from 6:30-8:30pm at the [Tech Cooperative](#). Meetings are informal and open to anyone interested in the Python language. We encourage lively discussion between programmers of all skill levels. Subscriptions are provided below to learn more about the group.

[Meetup](#) - Join us on Meetup for upcoming events.

[Twitter](#) - Follow us on Twitter for announcements.

[Slack](#) - Group discussions are in the [KnoxPy Slack channel](#).

[GitHub](#) - This website is hosted on the [Technology Cooperative's GitHub account](#).

[Email](#) - Questions, comments, and other correspondence can be directed to knoxpy@googlegroups.com.

The Technology Cooperative is located in downtown Knoxville, TN 37902. The nearest free parking is available at Sweet P's BBQ.

Resources

Resources related to the Python programming language are provided below. Links are organized by category such as books, conferences, courses, text editors and IDEs, podcasts, visualization tools, and other Python sites.

Effective Computation in Physics	PyCharm
Elegant SciPy	Spyder
Flask Web Development	Sublime Text
The Hitchhiker's Guide to Python	Visual Studio Code
Python for Data Science Handbook	Python Bytes
Two Scoops of Django	Talk Python to Me
DjangoCon	Bokeh
PyCascades	Matplotlib
PyCon	Mayavi
PyOhio	Plotly
PyTennessee	Anaconda
SciPy	Enthought Canopy
Learn Python 3	NumFOCUS
Introduction to Computer Science and Programming Using Python	pyOpenSci
Learn Python	Python
Introduction To Python Programming	Real Python

Next steps...

27

Attend more KnoxPy meetings! <https://knoxpy.org>

The screenshot shows a web browser window displaying the KnoxPy website. The title bar reads "knoxpy.org". The main content area has a blue header with the text "KnoxPy 🐍💻" and "Python users group in Knoxville, TN". Below the header is a large white section containing text about the group's meetings and a "Subscribe" button. Further down are links to Meetup, Twitter, Slack, GitHub, and Email.

Welcome Pythonistas! The KnoxPy group meets on the first Thursday of every month from 6:30-8:30pm at the [Technology Cooperative](#) in downtown Knoxville. Meetings are informal and open to anyone interested in the Python programming language. We encourage lively discussion, demonstrations, and presentations by programmers of all skill levels. Subscribe to our newsletter and visit the links provided below to learn more about the group.

[Subscribe](#)

Meetup - Join us on Meetup for upcoming events and to view our past events.

Twitter - Follow us on Twitter for announcements and group discussion.

Slack - Group discussions are in the KnoxDevs #python channel on Slack.

GitHub - This website is hosted on the KnoxPy GitHub.

Email - Questions, comments, and other feedback.

The Technology Cooperative is located at 127 West Jackson Avenue, Unit 103,