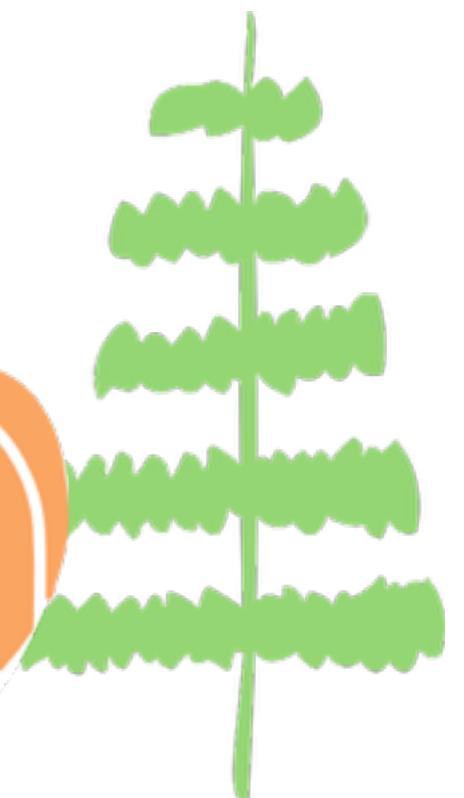
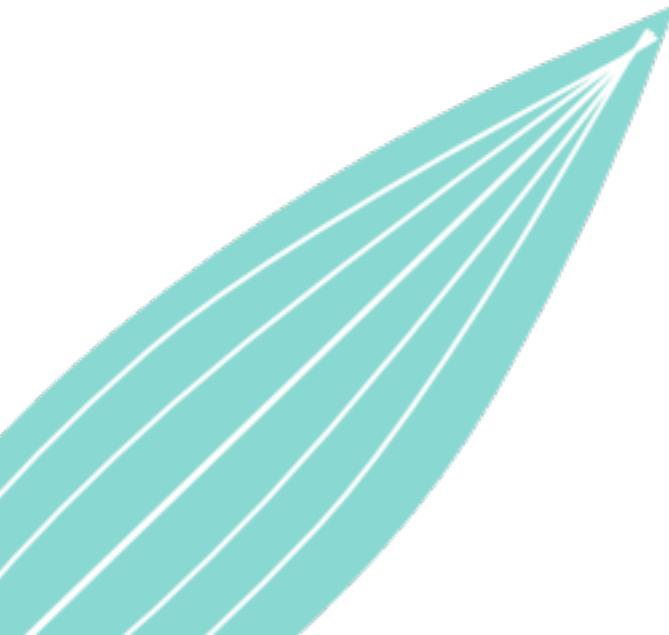


# Intro to Python: Conclusion

Stdlib, PyPI, and where to go from here



# Don't Reinvent the Wheel

You don't have to just reuse your own code.

You can also find and reuse code from:

- Python standard library (stdlib)
- Python Package Index (PyPI)

We'll go into these next.

# Python Standard Library

Also called the **stdlib**, it:

- Comes pre-installed with Python.
- Has modules for math, path handling, csv files, json, regular expressions, fractions, etc.

# StdLib Has Modules Like Random

**random module:**

- Has pseudo-random number generators.

You can import it as if it were any other .py file:

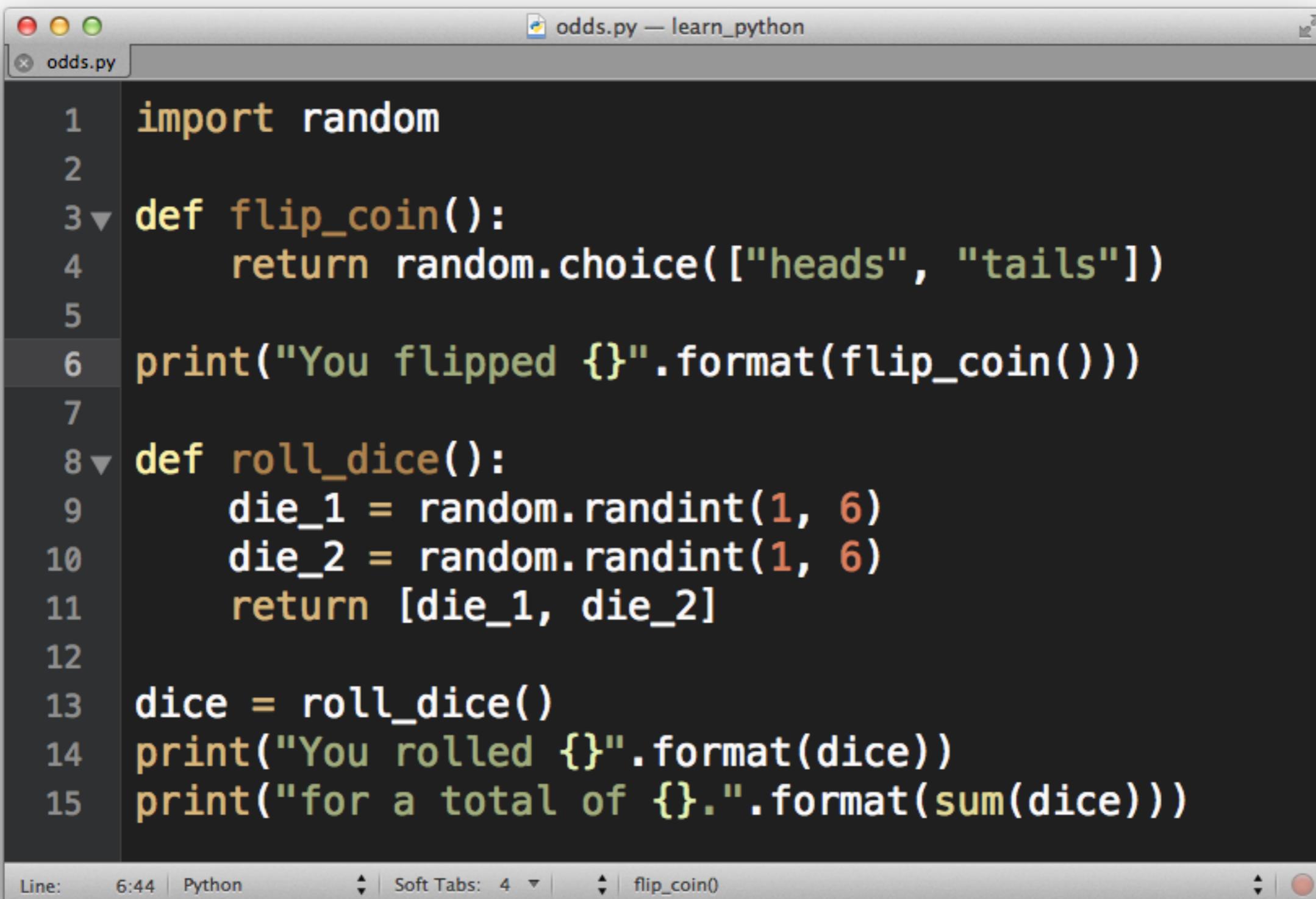
```
import random
```

# Example: Using Random

Create a file called **odds.py** and open it in your text editor.

Type in the code on the next slide...

# Example: Using Random



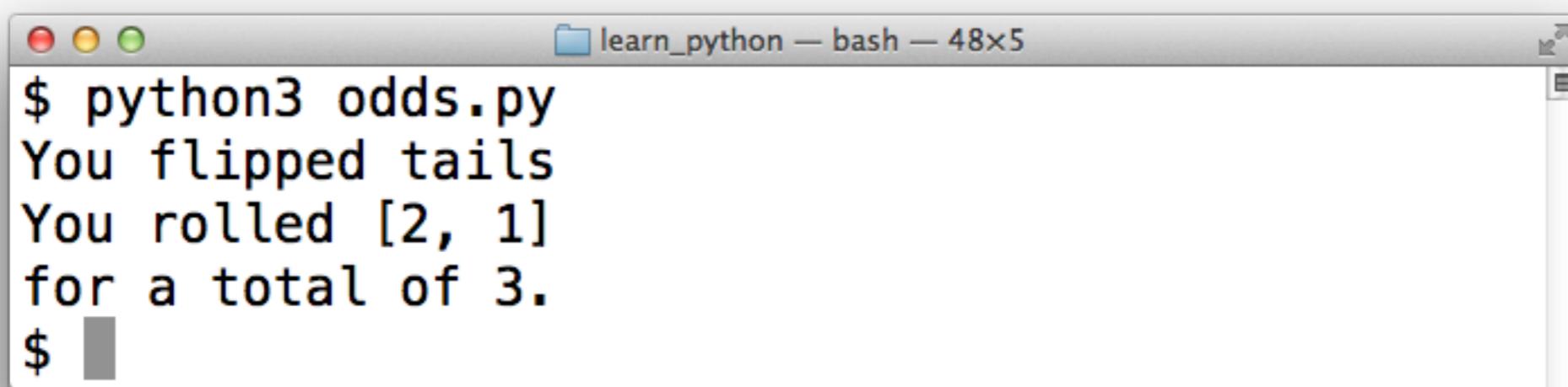
A screenshot of a Python code editor window titled "odds.py — learn\_python". The code editor displays the following Python script:

```
1 import random
2
3 def flip_coin():
4     return random.choice(["heads", "tails"])
5
6 print("You flipped {}".format(flip_coin()))
7
8 def roll_dice():
9     die_1 = random.randint(1, 6)
10    die_2 = random.randint(1, 6)
11    return [die_1, die_2]
12
13 dice = roll_dice()
14 print("You rolled {}".format(dice))
15 print("for a total of {}".format(sum(dice)))
```

The status bar at the bottom shows "Line: 6:44 | Python" and "Soft Tabs: 4". A cursor is visible on line 15.

# Example: Using Random

Run the program, as usual:



```
$ python3 odds.py
You flipped tails
You rolled [2, 1]
for a total of 3.
$
```

# StdLib is Pre-Installed

Just import a module and use it! No installation necessary.

The screenshot shows a web browser window with the title "The Python Standard Library". The address bar displays "Python Software Foundation [US] https://docs.python.org/3.4/library/". The page content is a hierarchical list of Python standard library modules. On the left, there's a sidebar with links to "Previous topic" (10. Full Grammar specification), "Next topic" (1. Introduction), "This Page" (Report a Bug, Show Source), and a "Quick search" field with a "Go" button. The main content area lists modules under several categories:

- 5.4. exception hierarchy
- 6. Text Processing Services
  - 6.1. string — Common string operations
  - 6.2. re — Regular expression operations
  - 6.3. difflib — Helpers for computing deltas
  - 6.4. textwrap — Text wrapping and filling
  - 6.5. unicodedata — Unicode Database
  - 6.6. stringprep — Internet String Preparation
  - 6.7. readline — GNU readline interface
  - 6.8. rlcompleter — Completion function for GNU readline
- 7. Binary Data Services
  - 7.1. struct — Interpret bytes as packed binary data
  - 7.2. codecs — Codec registry and base classes
- 8. Data Types
  - 8.1. datetime — Basic date and time types
  - 8.2. calendar — General calendar-related functions
  - 8.3. collections — Container datatypes
  - 8.4. collections.abc — Abstract Base Classes for Containers
  - 8.5. heapq — Heap queue algorithm
  - 8.6. bisect — Array bisection algorithm
  - 8.7. array — Efficient arrays of numeric values
  - 8.8. weakref — Weak references
  - 8.9. types — Dynamic type creation and names for built-in types
  - 8.10. copy — Shallow and deep copy operations
  - 8.11. pprint — Data pretty printer
  - 8.12. reprlib — Alternate repr() implementation
  - 8.13. enum — Support for enumerations
- 9. Numeric and Mathematical Modules
  - 9.1. numbers — Numeric abstract base classes
  - 9.2. math — Mathematical functions
  - 9.3. cmath — Mathematical functions for complex numbers
  - 9.4. decimal — Decimal fixed point and floating point arithmetic

# Python Package Index

Some packages in here are so much better than the stdlib.

The screenshot shows the PyPI (Python Package Index) website. The URL in the browser is <https://pypi.python.org/pypi>. The page features a navigation bar with links for "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", and "PyPI Developer Info". Below the navigation bar is a search bar and a "search" button. The main content area has a heading "PyPI - the Python Package Index" and a sub-heading "» Package Index". It includes sections for "Get Packages", "Package Authors", and "Infrastructure". A sidebar on the right provides links for "Not Logged In" (Login, Register, Lost Login?, Use OpenID), "Status" (Nothing to report), and "About" (JSON, OAuth, XML-RPC, HTTP interfaces). At the bottom, there is a table showing recently updated packages:

Updated	Package	Description
2014-09-19	<a href="#">induction 0.1</a>	A simple web framework based on asyncio.
2014-09-19	<a href="#">unicore-cms 0.2.1</a>	JSON based CMS for Universal Core
2014-09-19	<a href="#">PyUPC-EAN 2.6.2</a>	A barcode library/module for python.
2014-09-19	<a href="#">plonetheme.INTKmodern 0.1.0</a>	An installable Diazo theme for Plone 4
2014-09-19	<a href="#">mad2 0.1.12</a>	file metadata tagger
2014-09-19	<a href="#">rainbowstream 1.0.5</a>	A smart and nice Twitter client on terminal.

# Package vs. Module

A package is a folder containing:

- Useful modules
- Sometimes other packaging cruft
  - e.g. files to help release your package on PyPI

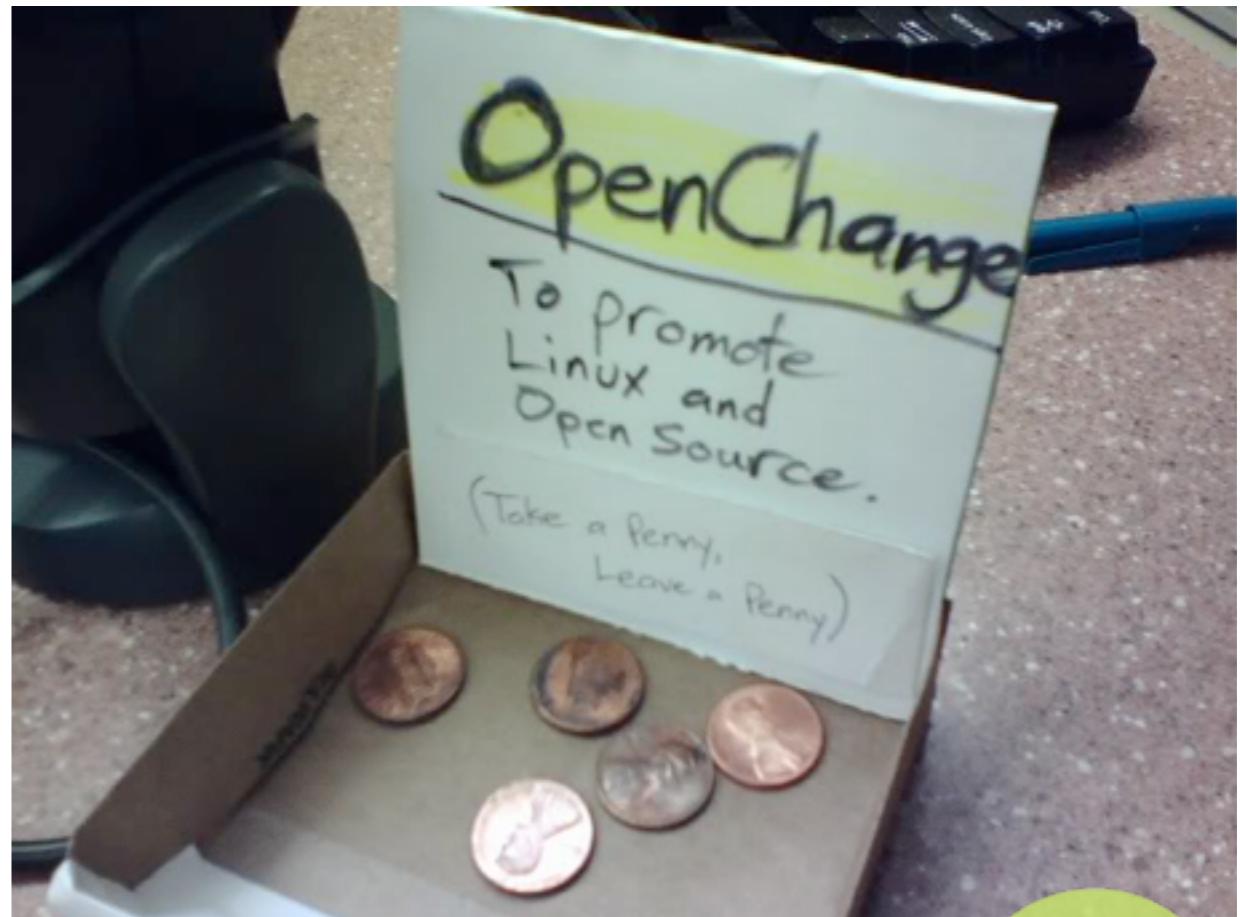
# FOSS

There's a lot of FOSS Python code available.

<http://ubuntuforums.org/showthread.php?t=549713>

## Free and Open Source Software

- Yay, free stuff!
- Treasure trove for projects
- Use it, give back too



[http://en.wikipedia.org/wiki/Free\\_and\\_open-source\\_software](http://en.wikipedia.org/wiki/Free_and_open-source_software)

# Package Example

The screenshot shows a web browser window with the following details:

- Title Bar:** Shows the title "binaryornot 0.3.0 : Python" and the URL "Python Software Foundation [US] https://pypi.python.org/pypi/binaryornot/0.3.0".
- Header:** Features the Python logo and navigation links like "search".
- Breadcrumbs:** "» Package Index > binaryornot > 0.3.0".
- Left Sidebar (PACKAGE INDEX):** Includes links for "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", and "PyPI Developer Info".
- Right Sidebar (Not Logged In):** Includes links for "Login", "Register", "Lost Login?", "Use OpenID", and "Status".
- Content Area:**
  - Section Header:** "binaryornot 0.3.0".
  - Description:** "Ultra-lightweight pure Python package to check if a file is binary or text."
  - Download Button:** "Download binaryornot-0.3.0.tar.gz".
  - Badges:** "pypi package 0.3.0", "build passing", "downloads 3.3k/month".
  - Text:** "Ultra-lightweight pure Python package to guess whether a file is binary or text, using a heuristic similar to Perl's `pp_fttext` and its analysis by @eliben."
  - List:** "• Free software: BSD license  
• Documentation: <http://binaryornot.readthedocs.org>".
  - Status Section:** "Status".
  - Text:** "It works, and I'm using this package in various places. But it doesn't cover all edge cases yet."
  - Text:** "The code could be improved. Pull requests welcome! As of now, it is based on these snippets, but that may change:"
  - List:** "• <http://stackoverflow.com/questions/898669/how-can-i-detect-if-a-file-is-binary-non-text-in-python>  
• <http://stackoverflow.com/questions/1446549/how-to-identify-binary-and-text-files-using-python>  
• <http://code.activestate.com/recipes/173220/>  
• <http://eli.thegreenplace.net/2011/10/19/perl-s-guess-if-file-is-text-or-binary-implemented-in-python/>
  - Features Section:** "Features".
  - Text:** "Has tests for these file types:"
  - List:** "• Text: .css, .json, .txt, .svg".

# Exercise: Explore a Package

Go to <https://pypi.python.org>:

- Search for *binaryornot*
  - or “binary or not”

# Exercise: Explore a Package

Study the package page:

The screenshot shows a web browser window with the following details:

- Title Bar:** Shows the title "binaryornot 0.3.0 : Python" and the URL "https://pypi.python.org/pypi/binaryornot/".
- Header:** Features the Python logo and navigation links like "search".
- Breadcrumbs:** "» Package Index > binaryornot > 0.3.0"
- Section Header:** "binaryornot 0.3.0"
- Description:** "Ultra-lightweight pure Python package to check if a file is binary or text."
- Buttons:** "Download binaryornot-0.3.0.tar.gz" and "pypi package 0.3.0 build passing downloads 3.3k/month".
- Text:** "Ultra-lightweight pure Python package to guess whether a file is binary or text, using a heuristic similar to Perl's `pp_fttext` and its analysis by @eliben."
- List:** "• Free software: BSD license  
• Documentation: <http://binaryornot.readthedocs.org>"
- Section Header:** "Status"
- Text:** "It works, and I'm using this package in various places. But it doesn't cover all edge cases yet."
- Text:** "The code could be improved. Pull requests welcome! As of now, it is based on these snippets, but that may change:"
- List:** "• <http://stackoverflow.com/questions/898669/how-can-i-detect-if-a-file-is-binary-non-text-in-python>  
• <http://stackoverflow.com/questions/1446549/how-to-identify-binary-and-text-files-using-python>  
• <http://code.activestate.com/recipes/173220/>  
• <http://eli.thegreenplace.net/2011/10/19/perl-s-guess-if-file-is-text-or-binary-implemented-in-python/>
- Section Header:** "Features"
- Text:** "Has tests for these file types:"
- Right Sidebar:** "Not Logged In" section with links: "Login", "Register", "Lost Login?", "Use OpenID", "Status", and "Nothing to report".

# Exercise: Explore a Package

Scroll down and find the package's Home Page:

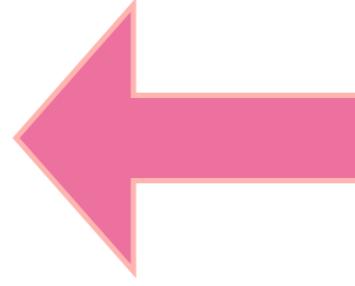
A screenshot of a web browser displaying the PyPI page for the `binaryornot` package. The browser window title is "binaryornot 0.3.0 : Python". The address bar shows the URL <https://pypi.python.org/pypi/binaryornot/>. The page content includes release notes for version 0.1.1 (2013-08-17) and 0.1.0 (2013-08-17), a file download table, download statistics, and package metadata.

File Type Py Version Uploaded on Size

<a href="#">binaryornot-0.3.0.tar.gz (md5)</a>	Source		2014-05-05	212KB
--	--------	--	------------	-------

**Downloads (All Versions):**

- 79 downloads in the last day
- 716 downloads in the last week
- 3372 downloads in the last month

**Author:** Audrey Roy 

**Home Page:** <https://github.com/audreyr/binaryornot>

**Keywords:** binaryornot

**License:** BSD

**Categories**

- Development Status :: 4 - Beta
- Intended Audience :: Developers
- License :: OSI Approved :: BSD License
- Natural Language :: English
- Programming Language :: Python :: 2
- Programming Language :: Python :: 2.6

# Exercise: Explore a Package

Now browse the code for the package:

The screenshot shows a web browser window displaying the GitHub repository page for the package `binaryornot`. The URL in the address bar is <https://github.com/audreyr/binaryornot>. The repository name is `audreyr / binaryornot`. The description is "Ultra-lightweight pure Python package to check if a file is binary or text." The repository has 63 commits, 3 branches, 3 releases, and 2 contributors. The master branch is selected. A list of recent commits is shown, including:

- Release 0.3.0 by `audreyr` on May 5, 5 months ago.
- Better PYTHONPATH handling when building Sphinx docs. by `audreyr` on a year ago.
- Read files in binary mode to simplify implementation by `audreyr` on a year ago.
- Ignore coverage output, profiling stats file. by `audreyr` on a year ago.
- Initial commit. by `audreyr` on a year ago.
- Add @vincentbernat to AUTHORS. by `audreyr` on 5 months ago.
- Initial commit. by `audreyr` on a year ago.
- Release 0.3.0. by `audreyr` on 5 months ago.

The right sidebar contains links for Code, Issues (0), Pull Requests (0), Wiki, Pulse, Graphs, and Settings. It also provides an SSH clone URL: `git@github.com:audreyr/binaryornot`.

# Exercise: Explore a Package

The actual Python code is in the *binaryornot/* directory:

- *check.py*
- *helpers.py*

Look at those files. They're just modules containing simple Python functions.

# Exercise: Explore a Package

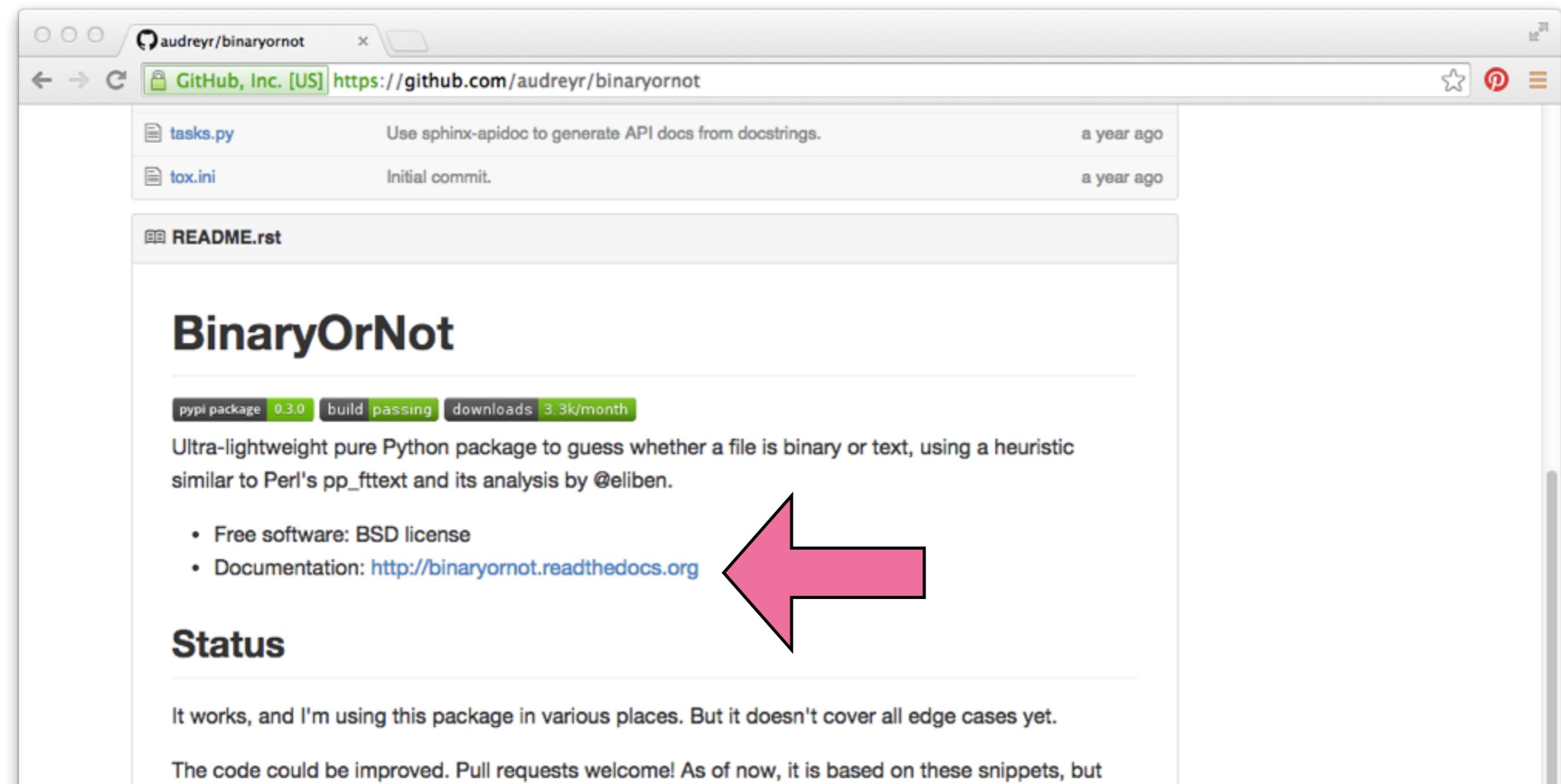
What about all the other files?

- `__init__.py` is needed for Python 2 compatibility
- `.rst` files, `docs/` are for documentation
- `setup.py`, `MANIFEST.in` is packaging cruft
- `tox`, `travis`, `tests/` are for testing

All this boilerplate can be auto-generated, don't worry

# Exercise: Explore a Package

Find the link to the package's documentation:



The screenshot shows a web browser window with the GitHub repository page for `audreyr/binaryornot`. The URL in the address bar is <https://github.com/audreyr/binaryornot>. The repository has two commits: `tasks.py` (a year ago) and `tox.ini` (a year ago). The `README.rst` file is open, displaying the following content:

## BinaryOrNot

pypi package 0.3.0 build passing downloads 3.3k/month

Ultra-lightweight pure Python package to guess whether a file is binary or text, using a heuristic similar to Perl's `pp_fttext` and its analysis by @eliben.

- Free software: BSD license
- Documentation: <http://binaryornot.readthedocs.org>

## Status

It works, and I'm using this package in various places. But it doesn't cover all edge cases yet.

The code could be improved. Pull requests welcome! As of now, it is based on these snippets, but

A large pink arrow points from the word "Documentation" in the list above to the URL <http://binaryornot.readthedocs.org>.

# Exercise: Explore a Package

The documentation for a package teaches you how to install and use it.

A screenshot of a web browser window showing the documentation for the `BinaryOrNot` package. The URL in the address bar is `binaryornot.readthedocs.org/en/latest/`. The page title is "Welcome to BinaryOrNot's documentation!". On the left sidebar, there are sections for "Project Versions" (latest), "RTD Search" (with a search input field and "Go" button), "Table Of Contents" (listing "Welcome to BinaryOrNot's doc" and "Indices and tables"), and "Next topic" (listing "BinaryOrNot"). The main content area contains a heading "Welcome to BinaryOrNot's documentation!" followed by a "Contents:" section with a hierarchical list of topics:

- BinaryOrNot
  - Status
  - Features
  - Why?
  - Credits
- Installation
- Quickstart
- Contributing
  - Types of Contributions
  - Get Started!
  - Pull Request Guidelines
  - Tips
- Credits
  - Development Lead

# Installing Packages

Common package installation tools:

- pip
- anaconda (“conda”)
- easy\_install

Always read a package’s docs for installation advice, though.

# Creating new Packages

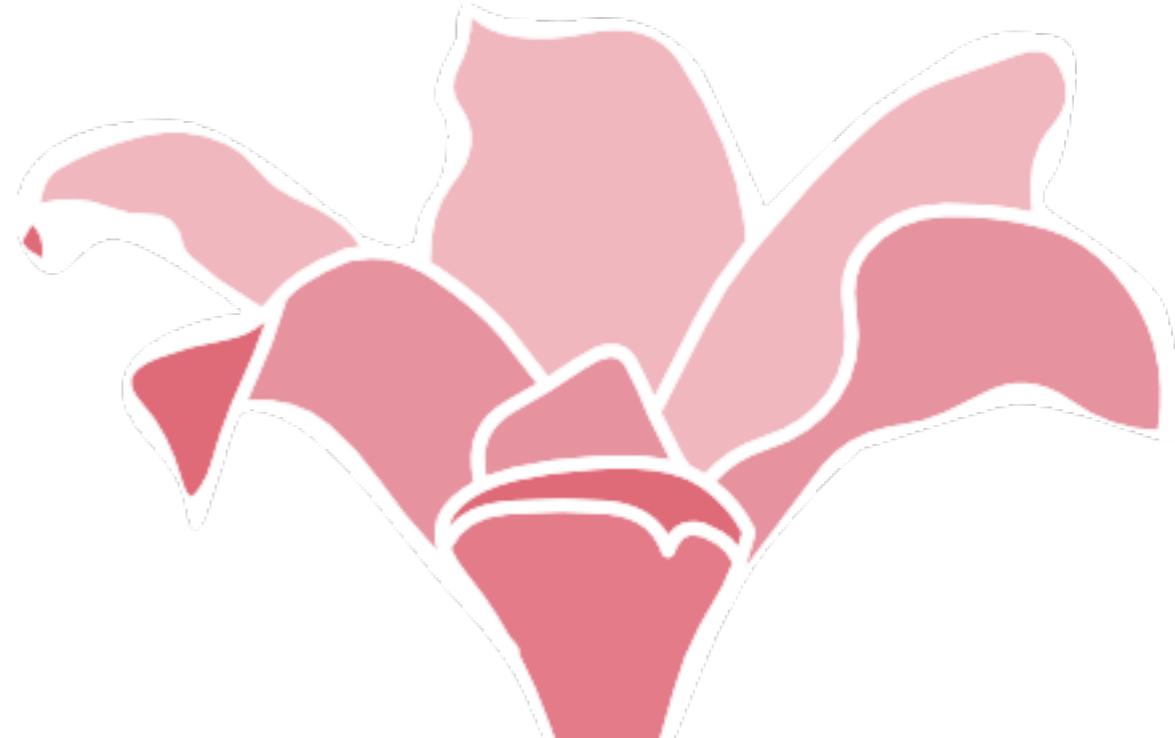
If you continue with Python coding, you'll want to create and release your own packages

- Look up **cookiecutter**
  - Tool for generating the package boilerplate files



# Where to go from here

Your next adventures in Python coding



# Review on your Own! Study!

Start with the official Python docs & tutorial:

The screenshot shows a web browser window displaying the Python 3.4.1 documentation. The title bar reads "Overview — Python 3.4.1 c x". The address bar shows "Python Software Foundation [US] https://docs.python.org/3/". The page content includes a sidebar with links for "Download", "Docs for other versions" (listing Python 2.7, 3.3, 3.5, and Old versions), "Other resources" (PEP Index, Beginner's Guide, Book List, Audio/Visual Talks), and a "Quick search" bar. The main content area features the title "Python 3.4.1 documentation" and a welcome message: "Welcome! This is the documentation for Python 3.4.1, last updated Sep 18, 2014." It lists several documentation sections: "Parts of the documentation:", "What's new in Python 3.4?", "Tutorial", "Library Reference", "Language Reference", "Python Setup and Usage", "Python HOWTOs", "Installing Python Modules", "Distributing Python Modules", "Extending and Embedding", "Python/C API", and "FAQs". Each section has a brief description below it.

Download

Download these documents

Docs for other versions

Python 2.7 (stable)  
Python 3.3 (stable)  
Python 3.5 (in development)  
Old versions

Other resources

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

Quick search

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

Go

Overview — Python 3.4.1 c x

Python Software Foundation [US] <https://docs.python.org/3/>

Python » 3.4.1 Documentation » modules | index

## Python 3.4.1 documentation

Welcome! This is the documentation for Python 3.4.1, last updated Sep 18, 2014.

Parts of the documentation:

**What's new in Python 3.4?**  
*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*

**Installing Python Modules**  
*installing from the Python Package Index & other sources*

**Distributing Python Modules**  
*publishing modules for installation by others*

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

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

**FAQs**  
*frequently asked questions (with answers!)*

Indices and tables:

# Free Online University

Normally something like this would cost \$\$\$, but we are lucky to get it for free:

**coursera** | Global Partners

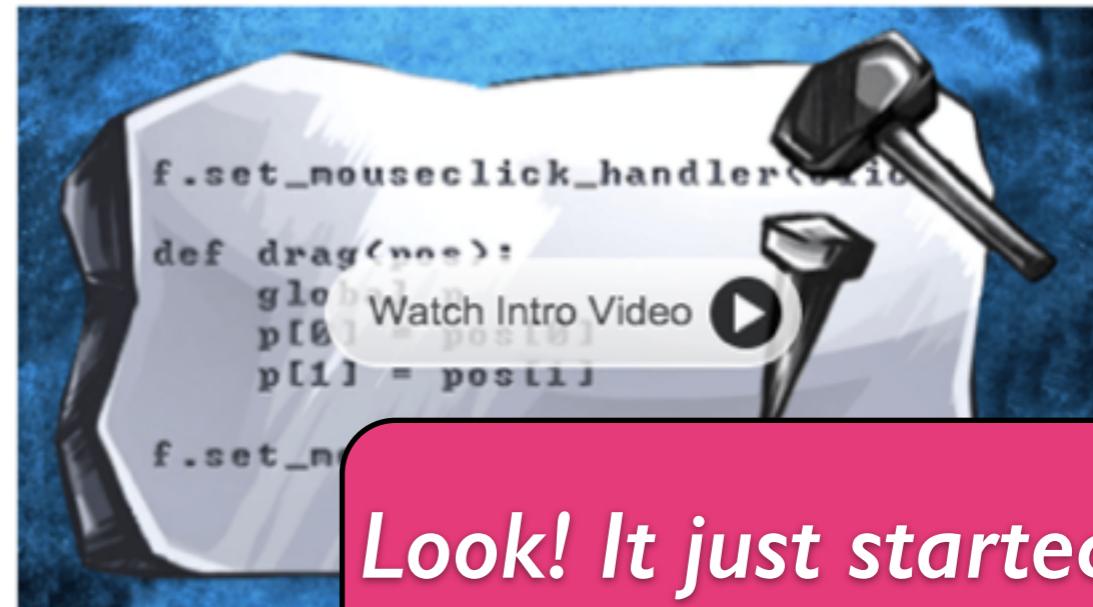
Courses Specializations Institutions About ▾ | Sign In Sign Up



## An Introduction to Interactive Programming in Python

Part of the "Fundamentals of Computing" Specialization »

This course is designed to be a fun introduction to the basics of programming in Python. Our main focus will be on building simple interactive games such as Pong, Blackjack and Asteroids.



**Look! It just started!  
Sign up!**

### About the Course

This course is designed to help students with very little or no computing background learn the basics of building simple interactive applications. Our language of choice,

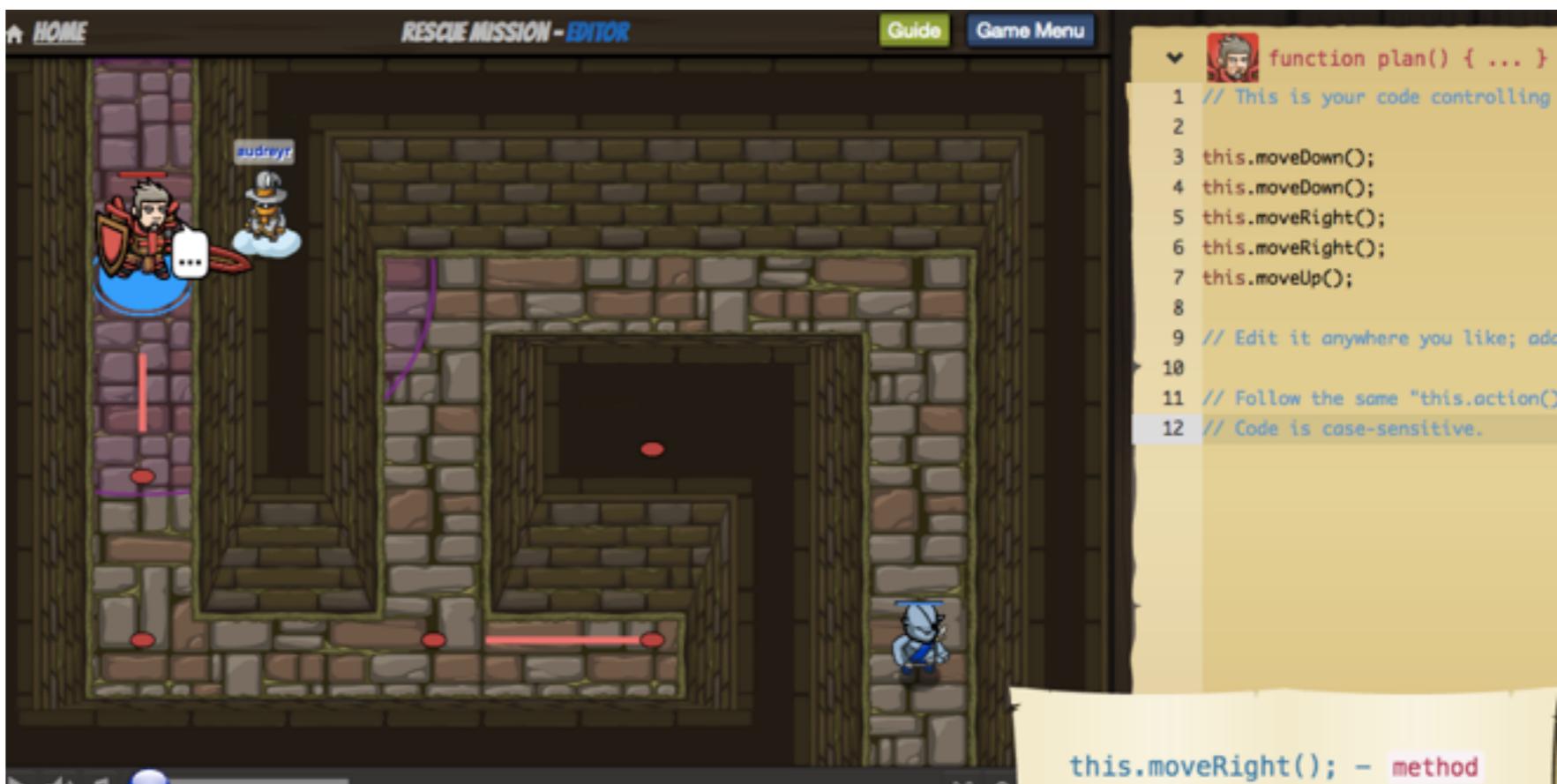
### Sessions

Sep 15th 2014 - Nov 16th 2014

# Interactive Python Walkthroughs/Games

Websites that turn learning Python into an interactive experience or even a puzzle game:

- [codewars.com/?language=python](https://www.codewars.com/?language=python)
- [codecademy.com/en/tracks/python](https://www.codecademy.com/en/tracks/python)
- [codecombat.com](https://www.codecombat.com)
- [pythontutor.com](https://pythontutor.com)



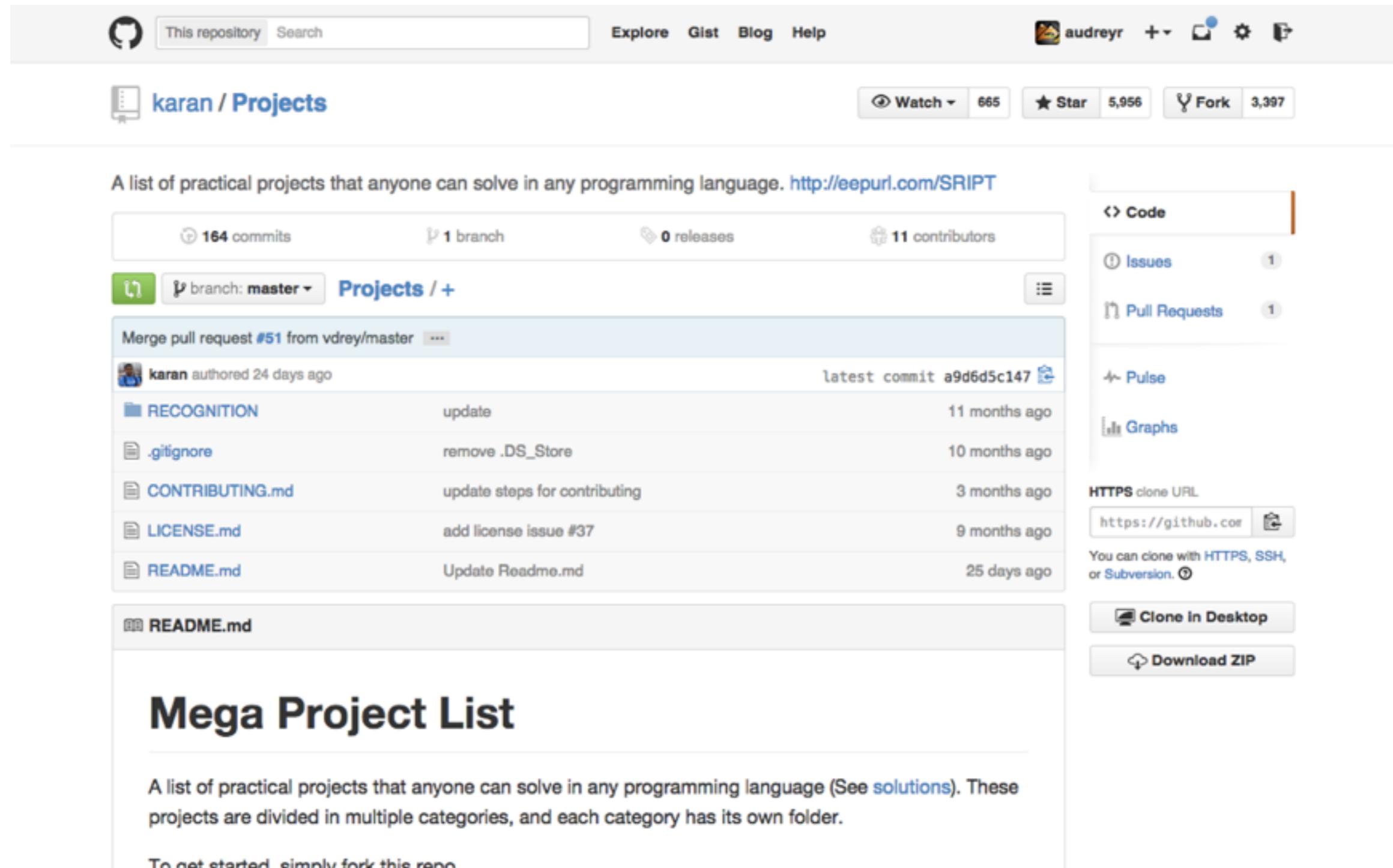
# Books (“Old Technology”)

There's still nothing quite like learning from a book the old-fashioned way.



# Your First Coding Project

Stuck without ideas? See Karan's Mega Project List:  
<https://github.com/karan/Projects>



The screenshot shows the GitHub repository page for 'karan / Projects'. The repository has 164 commits, 1 branch, 0 releases, and 11 contributors. It contains files like RECOGNITION, .gitignore, CONTRIBUTING.md, LICENSE.md, and README.md. The README.md file is titled 'Mega Project List' and describes it as a collection of practical projects anyone can solve in any programming language. The repository has 665 stars and 3,397 forks.

A list of practical projects that anyone can solve in any programming language. <http://eepurl.com/SRIPT>

164 commits 1 branch 0 releases 11 contributors

branch: master Projects / +

Merge pull request #51 from vdrey/master ...  
karan authored 24 days ago latest commit a9d6d5c147

File	Commit Message	Date
RECOGNITION	update	11 months ago
.gitignore	remove .DS_Store	10 months ago
CONTRIBUTING.md	update steps for contributing	3 months ago
LICENSE.md	add license issue #37	9 months ago
README.md	Update Readme.md	25 days ago

README.md

## Mega Project List

A list of practical projects that anyone can solve in any programming language (See [solutions](#)). These projects are divided in multiple categories, and each category has its own folder.

To get started, simply fork this repo.

Code Issues Pull Requests Pulse Graphs

HTTPS clone URL  
<https://github.com>

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

Clone In Desktop Download ZIP

# Ask Questions

Shy? Use a pseudonym. Many of us do that.

The screenshot shows the Stack Overflow homepage with the search term "[python]" entered in the search bar. The main content area displays tagged questions for Python, with two examples shown:

**Tagged Questions**

**What does the yield keyword do in Python?**  
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

**What is a metaclass in Python?**  
What are metaclasses? What do you use them for?  
python oop metaclass python-datamodel

**Looking for a job?**

# On the Internet, You Can Be Anonymous

Take advantage of it:

- Ask for help liberally!
- Put your code online, e.g. GitHub
- Try putting a package on PyPI and see what happens

# On the Internet, Nobody Knows You're a New Coder

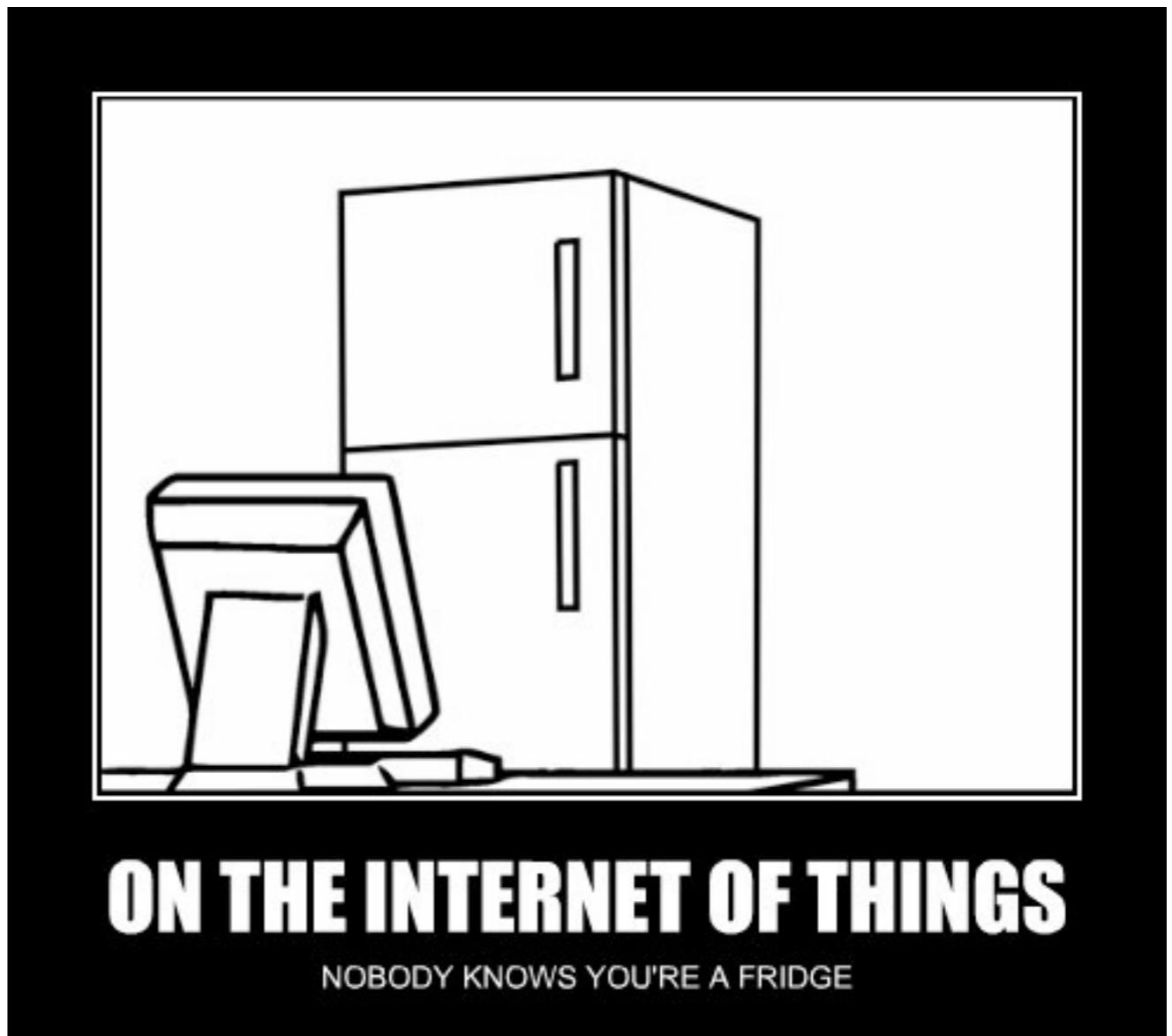


*"On the Internet, nobody knows you're a dog."*

For all they know,  
you could be a dog.

# On the Internet, Nobody Knows You're a New Coder

Or even a fridge.



**ON THE INTERNET OF THINGS**

NOBODY KNOWS YOU'RE A FRIDGE

# Attend Coding Meetups

Gotta catch ‘em all (free education, meet peers)

The screenshot shows the Meetup.com homepage with a blue header bar. At the top left is the Meetup logo. To its right are two buttons: "Find a Meetup Group" and "Start a Meetup Group". On the far right of the header are icons for a profile picture (a woman), messaging, notifications, and more options. Below the header, a large blue section titled "Your Next Meetup" displays an event for "Inland Empire Pyladies" on "SEP 20" for an "Intro to Python Workshop (joint event)" on Saturday, September 20 at 9:30 AM, with 7 comments. To the right of this main area are two circular statistics: one for "Meetups in your groups" (81) and another for "Meetups with friends" (57). At the bottom of the page is a dark navigation bar with categories: Programming, within 25 miles of San Diego, CA, Groups, and Calendar.

SATURDAY, SEP 20

This screenshot shows a detailed view of the "Intro to Python Workshop (joint event)" listed under the "SATURDAY, SEP 20" heading. It includes the time (9:30 AM), the organizer (Inland Empire Pyladies), a checked-in status (Yes), the event name, and the number of attendees (4 Pythonistas going). To the right of the event details is a sidebar with a profile picture and the text "1 friend".

- All Meetups
- My Meetups & suggestions
- My Meetups
- I'm going

# Join the Nearest PUG

PUG means Python User Group



PyLadies of San Francisco

<http://www.meetup.com/PyLadiesSF/>

Bay Area Python Interest Group

<http://www.meetup.com/BayPIGgies/>

# Attend a Conference!

PyCon 2016 — Portland OR, Spring <http://us.pycon.org/>

DjangoCon 2015 — Austin TX, September 6-11 <https://2015.djangoproject.us/>

PyData Seattle 2015 — Seattle WA, July 24-26 <http://pydata.org/>

SciPy 2015 — Austin TX, July 6-12 <http://scipy2015.scipy.org/>

# Keep on Coding!

We are all unpaid volunteers, here because we want to help the community.

- So take the time to keep learning.
- You get as much out of this workshop as you put in.

Thank you!

# Quick Survey

Please help us by doing this survey **right now**:

- <https://www.surveymonkey.com/s/S9WYFXQ>

Your feedback helps us plan future community workshops and events.