

*Come to the Dark Side!
We have a whole bunch of Cookiecutters!*

Raphael Pierzina @hackebrot



EUROPYTHON
2015 Bilbao, 20-26 July

Raphael Pierzina

- Background in **3D Animation**
- Core Developer of **Cookiecutter**
- Volunteer for Adopt **Pytest** Month
- Technical Reviewer on a **Kivy** Book





/ hackebrot



@hackebrot

“What is thy bidding, my master?”

–Darth Vader (The Empire Strikes Back)



Guido van Rossum @gvanrossum · 14 Apr 2014

Can we rename newbies to puppies?

82 73



Raphael Pierzina @hackebrot · 14 Apr 2014

@gvanrossum I like 'padawan' best!



Guido van Rossum @gvanrossum · 15 Apr 2014

.@hackebrot Hm, I think people will think the bar for a padawan too high. Anyone can be a puppy though.

RETWEETS

3

FAVORITES

4



8:02 AM - 15 Apr 2014 · Details

[Hide conversation](#)



A command-line utility that creates projects from cookiecutters (project templates). E.g. Python package projects, jQuery plugin projects.

Artwork by Audrey Roy Greenfeld

GitHub

 [audreyr](#) / [cookiecutter](#)


 Unwatch ▾ 64

 Unstar 1,942

 Fork 242

A command-line utility that creates projects from cookiecutters (project templates). E.g. Python package projects, jQuery plugin projects.


 1,105 commits

 13 branches

 17 releases

 67 contributors



 branch: **master** ▾

[cookiecutter](#) / +



Add changes of [#470](#) and [#471](#) to changelog in HISTORY.rst



hackebrot authored 7 minutes ago

latest commit [c729de4591](#) 

 **Code**

 [Issues](#) 77

 [Pull requests](#) 45

 [Wiki](#)

cookiecutter 1.0.0 : Python ...

Python Software Foundation (US) <https://pypi.python.org/pypi/cookiecutter/1.0.0>

python™

» Package Index > cookiecutter > 1.0.0

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 »

cookiecutter 1.0.0

A command-line utility that creates projects from project templates, e.g. creating a Python package project from a Python package project template.

Downloads ↓

pypi package 1.0.0 build passing BUILD PASSING Codecov 96% GITTER JOINCHAT → docs latest

A command-line utility that creates projects from **cookiecutters** (project templates), e.g. creating a Python package project from a Python package project template.

- Documentation: <http://cookiecutter.rtdf.org>
- GitHub: <https://github.com/audreyr/cookiecutter>
- Free software: BSD license
- PyPI: <https://pypi.python.org/pypi/cookiecutter>



COOKIECUTTER

Features

Did someone say features?

- Cross-platform: Windows, Mac, and Linux are officially supported.
- Works with Python 2.6, 2.7, 3.3, 3.4, and PyPy. (But you don't have to know/write Python code to use Cookiecutter.)

Not Logged In

- [Login](#)
- [Register](#)
- [Lost Login?](#)
- [Use OpenID](#)  

Status

[Nothing to report](#)

```
$ [sudo] pip install cookiecutter
```


Features

- Cross-platform: Windows, Mac, and Linux are officially supported.
- Free software distributed under the terms of the BSD license
- Works with Python 2.7, 3.3, 3.4, and PyPy.
- Unicode support for named platforms and Python versions
- Project templates can be in any programming language or markup format

Technologies

- 100% of templating is done with Jinja2
- Support for Git and Mercurial
- Sphinx Docs hosted on Read the Docs
- Running Tox and Pytest **(at 96% test coverage)**
- CLI and user prompting via Click

Community

The core committer team is [@audreyr](#), [@pydanny](#), [@michaeljoseph](#), [@pfmoore](#), and [@hackebrot](#).

We welcome you and invite you to participate.

Up to date 61 contributors and template authors in total.



Audrey Roy Greenfeld @audreyr · Jul 9
The aftermath of me and @pydanny recording our @djangobirthday tribute video.

6 41



“After EuroPython 2013, Danny and I both came away from the conference with a renewed spirit and enthusiasm to sprint on open source.”

–Audrey Roy Greenfeld

“I had a lot of ideas for packaging various snippets of my existing code as new Python libraries. But I found that I was doing a lot of repetitive work every time I created a new Python package. ”

–Audrey Roy Greenfeld

```
cookiecutter — vim — 117x48
vim | bash +
1 commit 97b01f8d4700ec7fe8f25a940a53c644cd882641
2 Author: Raphael Pierzina <raphael@hackebrot.de>
3 Date: Tue May 6 08:57:59 2014 +0200
4
5 Add template for the kivy python-framework to README.rst
6
7 commit ae535222049b30f18ed4e6f423853ad9ccb74965
8 Author: Raphael Pierzina <raphael@hackebrot.de>
9 Date: Mon Nov 10 21:08:00 2014 +0100
10
11 Create initial conftest.py for global fixtures
12
13 commit 952289a08784bc0ee86dee6239025397395f7033
14 Author: Raphael Pierzina <raphael@hackebrot.de>
15 Date: Mon Nov 10 21:18:26 2014 +0100
16
17 Change doc str to sphinx format
18
19 commit 55bf12c80fe14b317fdc27d74c27642a144d4530
20 Author: Raphael Pierzina <raphael@hackebrot.de>
21 Date: Mon Nov 10 21:23:58 2014 +0100
22
23 Create new test module to convert the first test
24
25 commit 55ef2990cba3d038e1d7fc9e2366b75f877aa896
26 Author: Raphael Pierzina <raphael@hackebrot.de>
27 Date: Mon Nov 10 21:40:31 2014 +0100
28
```

```
$ git log --author="Raphael Pierzina" --reverse
```

You underestimate the power of
a good template

Creating a Cookiecutter from Scratch

Kivy 1.9.0 released!
New SDL2 provider, ffpyslayer, EffectWidget, FocusBehavior
2486 commits, 540 pull requests merged
104 commits for SDL2, 96 contributors.
Download Now!

Progress Bar
Switch

Monoline textinput
Multiline textinput
Second line

R 45 G 85 B 85 A 255 H 127 S 119 V 80 X #2d5555ff

Name
> bin
> boot
> data
> debian
> dev
> etc
> home
> lib
> lib32
> lib64
bin
debian

Kivy - Open source Python library for rapid development of applications that make use of innovative user interfaces, such as multi-touch apps.

<http://kivy.org/#home>

Create Template Root

```
$ git clone https://github.com/hackebrot/cookieozer.git
```

```
$ cd cookieozer
```

```
$ tree -a -L 1
```

```
├── .git
├── .gitignore
├── LICENSE
└── README.md
```

Python Package

```
$ mkdir -p {{cookiecutter.repo_name}}/{{cookiecutter.repo_name}}
```

```
$ cd $_
```

```
$ touch __init__.py
```

```
$ touch {{cookiecutter.repo_name}}.py
```

```
$ touch {{cookiecutter.app_class_name}}.kv
```

```
$ touch main.py
```

__init__.py

```
# -*- coding: utf-8 -*-
```

```
__author__ = '{{cookiecutter.full_name}}'  
__email__ = '{{cookiecutter.email}}'  
__version__ = '{{cookiecutter.version}}'
```

{{cookiecutter.repo_name}}.py

```
# -*- coding: utf-8 -*-

"""
{{cookiecutter.repo_name}}
=====

The root of :class:`{{cookiecutter.app_class_name}}` is created from the kv file.
"""

import kivy
kivy.require({{cookiecutter.kivy_version}})

from kivy.app import App

class {{cookiecutter.app_class_name}}(App):
    """Basic Kivy App with a user defined title."""

    title = '{{cookiecutter.app_title}}'

    def build(self):
        return self.root
```

main.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from {{cookiecutter.repo_name}} import {{cookiecutter.app_class_name}}

def main():
    {{cookiecutter.app_class_name}}().run()

if __name__ == '__main__':
    main()
```

`{{cookiecutter.app_class_name}}.kv`

`#:kivy {{cookiecutter.kivy_version}}`

Button:

`text: ' {{cookiecutter.app_title}} '`

cookiecutter.json

```
$ cd -
```

```
$ touch cookiecutter.json
```


cookiecutter.json

```
{  
  "full_name": "Raphael Pierzina",  
  "email": "raphael@hackebrot.de",  
  "app_title": "Hello World",  
  "app_class_name": "HelloWorldApp",  
  "repo_name": "helloworld",  
  "version": "0.1.0",  
  "kivy_version": "1.8.0"  
}
```

The ability to destroy a planet is
insignificant next to the power of
Cookiecutter.

Advanced Usage

Templates in Variables

```
{  
  "full_name": "Raphael Pierzina",  
  "email": "raphael@hackebrot.de",  
  "app_title": "Hello World",  
  "app_class_name": "{{cookiecutter.app_title|replace(' ', '')}}App",  
  "repo_name": "{{cookiecutter.app_title|replace(' ', '')|lower}}",  
  "version": "0.1.0",  
  "kivy_version": "1.8.0"  
}
```

Hooks

```
$ mkdir hooks
```

```
$ touch $_/post_gen_project.py
```

post_gen_project.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

"""Rename the generated kv file to be compatible with the original kivy kv file
detection of `App.load_kv()``."""

import os

package_dir = '{{cookiecutter.repo_name}}'
old_kv_file = os.path.join(package_dir, '{{cookiecutter.app_class_name}}.kv')

lower_app_class_name = '{{cookiecutter.app_class_name}}'.lower()
if (lower_app_class_name.endswith('app')):
    lower_app_class_name = lower_app_class_name[:-3]
new_kv_file = os.path.join(package_dir, '{}.kv'.format(lower_app_class_name))

os.rename(old_kv_file, new_kv_file)
```

Current Layout

```
$ tree
```

```
├── cookiecutter.json  
├── {{cookiecutter.repo_name}}  
│   ├── {{cookiecutter.repo_name}}  
│   │   ├── {{cookiecutter.app_class_name}}.kv  
│   │   ├── {{cookiecutter.repo_name}}.py  
│   │   ├── __init__.py  
│   │   └── main.py  
│   ├── LICENSE  
│   └── README.rst  
├── hooks  
│   └── post_gen_project.py  
├── LICENSE  
└── README.rst
```

Test Suite

```
$ mkdir \{\{cookiecutter.repo_name\}\}/tests  
$ cd $_  
  
$ touch conftest.py  
$ touch test_{\{cookiecutter.repo_name\}}.py
```

conftest.py

```
# -*- coding: utf-8 -*-
```

```
import pytest
```

```
@pytest.fixture(scope="session")
```

```
def app(request):
```

```
    """Uses the InteractiveLauncher to provide access to an app instance.
```

```
    The finalizer stops the launcher once the tests are finished.
```

```
    Returns:
```

```
        :class:`{{cookiecutter.app_class_name}}`: App instance
```

```
    """
```

```
from kivy.interactive import InteractiveLauncher
```

```
from {{cookiecutter.repo_name}}.{{cookiecutter.repo_name}} import {{cookiecutter.app_class_name}}
```


conftest.py

```
launcher = InteractiveLauncher({{cookiecutter.app_class_name}}())

def stop_launcher():
    launcher.safeOut()
    launcher.stop()

request.addfinalizer(stop_launcher)

launcher.run()
launcher.safeIn()
return launcher.app
```

test_{{cookiecutter.repo_name}}.py

```
# -*- coding: utf-8 -*-
```

```
def test_app_title(app):
```

```
    """Simply tests if the default app title meets the expectations.
```

```
    Args:
```

```
        app (:class:`{{cookiecutter.app_class_name}}`): Default app instance
```

```
    Raises:
```

```
        AssertionError: If the title does not match
```

```
    """
```

```
    assert app.title == '{{cookiecutter.app_title}}'
```

Setup

```
$ touch \{\{cookiecutter.repo_name\}\}/setup.py
```

Resources on Python Packaging

<https://packaging.python.org/en/latest/index.html>

<http://www.jeffknupp.com/blog/2013/08/16/open-sourcing-a-python-project-the-right-way/>

<http://blog.ionelmc.ro/2014/05/25/python-packaging/>

...

Sphinx Documentation

```
$ cd -
```

```
$ mkdir \{\{cookiecutter.repo_name\}\}/docs
```

```
$ cd $_
```

```
$ sphinx-quickstart
```

Sphinx Quickstart

- > Root path for the documentation [.]<ENTER>
- > Separate source and build directories (y/n) [n]: y
- > Name prefix for templates and static dir [_]: <ENTER>

- > Project name: **{{cookiecutter.app_title}}**
- > Author name(s): **{{cookiecutter.full_name}}**
- > Project version: **{{cookiecutter.version}}**
- > Project release [**{{cookiecutter.version}}**]: <ENTER>

- > Source file suffix [.rst]: <ENTER>

User Config

```
$ touch ~/.cookiecuttrc
```

~/cookiecutterrc

```
default_context:
```

```
  full_name: "Raphael Pierzina"
```

```
  email: "raphael@hackebrot.de"
```

```
  github_username: "hackebrot"
```

```
abbreviations:
```

```
  cookiedozer: https://github.com/hackebrot/cookiedozer.git
```

```
  gh: https://github.com/{0}.git
```


Showcase User Config

```
$ cookiecutter gh:audreyr/cookiecutter-pypackage
```

```
Cloning into 'cookiecutter-pypackage'...
```

```
remote: Counting objects: 505, done.
```

```
remote: Total 505 (delta 0), reused 0 (delta 0), pack-reused 505
```

```
Receiving objects: 100% (505/505), 77.43 KiB | 0 bytes/s, done.
```

```
Resolving deltas: 100% (265/265), done.
```

```
Checking connectivity... done.
```

```
full_name [Raphael Pierzina]:
```

```
email [raphael@hackebrot.de]:
```

```
github_username [hackebrot]:
```

```
project_name [Python Boilerplate]: Test User Config
```

```
repo_name [boilerplate]:
```

```
project_short_description [Python Boilerplate contains all the boilerplate you need to  
create a Python package.]:
```

```
release_date [2015-01-11]:
```

```
year [2015]:
```

```
version [0.1.0]:
```

“Bring my shuttle.”

–Darth Vader (The Empire Strikes Back)

Choices in Context

```
{  
  "full_name": "Raphael Pierzina",  
  "email": "raphael@hackebrot.de",  
  "github_username": "hackebrot",  
  "project_name": "Kivy Project",  
  "minimum_kivy_version": "1.8.0",  
  "orientation": ["all", "landscape", "portrait"]  
}
```

Choice Prompt

Select orientation:

1 - all

2 - landscape

3 - portrait

Choose from 1, 2, 3 [1]: 2

Click Prompts

```
def read_user_choice(var_name, options):  
    """Prompt the user to choose from several options for the given variable.  
  
    The first item will be returned if no input happens.  
  
    :param str var_name: Variable as specified in the context  
    :param list options: Sequence of options that are available to select from  
    :return: Exactly one item of ``options`` that has been chosen by the user  
    """  
  
    # Please see http://click.pocoo.org/4/api/#click.prompt  
    if not isinstance(options, list):  
        raise TypeError  
  
    if not options:  
        raise ValueError
```

Click Prompts

```
choice_map = OrderedDict(
    ('{}'.format(i), value) for i, value in enumerate(options, 1)
)
choices = choice_map.keys()
default = '1'

choice_lines = ['{} - {}'.format(*c) for c in choice_map.items()]
prompt = '\n'.join(
    'Select {}:' .format(var_name),
    '\n'.join(choice_lines),
    'Choose from {}'.format(', '.join(choices))
)

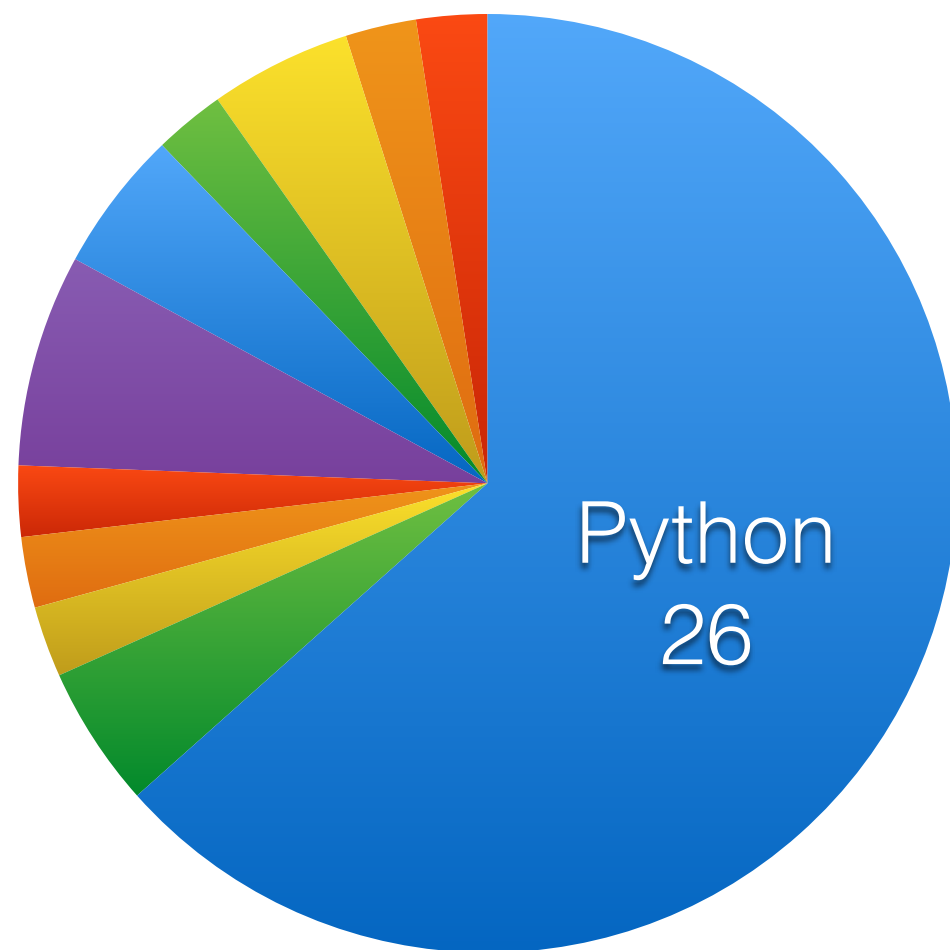
user_choice = click.prompt(
    prompt, type=click.Choice(choices), default=default
)
return choice_map[user_choice]
```

Join us, and together we can rule the
galaxy as Cookiecutters.

Template Contributions


Templates

41 Templates



- Python
- C++
- Common Lisp
- LaTeX/XeTeX
- HTML
- 6502 Assembly
- C
- C#
- JS
- Berkshelf-Vagrant
- Scala



cookiecutter-django

 [pydanny](#) / [cookiecutter-django](#)



[Watch](#) 61 [Unstar](#) 678 [Fork](#) 280

A cookiecutter template for creating Django projects quickly.

Python 50.9% **HTML 26.0%** **Makefile 7.3%** **Batchfile 6.6%** **Shell 4.6%** **JavaScript 3.0%** **CSS 1.6%**

 branch: **master** [cookiecutter-django](#) / + 

update Django version to 1.8.3 ...

 **luzfcb** authored 4 days ago latest commit [7837c4e099](#) 

[Code](#)

[Issues](#) 25

[Pull requests](#) 6

[Wiki](#)

Features

- For Django 1.8
- Twitter [Bootstrap 3](#)
- [AngularJS](#)
- [12-Factor](#) based settings via [django-environ](#)
- Optimized development and production settings
- Registration via [django-allauth](#)
- [Procfile](#) for deploying to Heroku
- Grunt build for compass and livereload
- Basic e-mail configurations for send emails via [Mailgun](#)
- Media storage using Amazon S3
- Serve static files from Amazon S3 or [Whitenoise](#) (optional)

cookiecutter-pylibrary



ionelmc / **cookiecutter-pylibrary**

Watch 8

Unstar 86

Fork 23

Enhanced cookiecutter template for Python libraries.

Python 71.1%

PowerShell 18.2%

Batchfile 7.1%

Shell 3.6%



branch: master

cookiecutter-pylibrary / +



Strip various sections and markup that would be unused with the given... ⋮



ionelmc authored a day ago

latest commit 2916d73125

Code

Issues 5

Pull requests 0

Pulse

Features

This is an "all inclusive" sort of template.

- BSD 2-clause license.
- [Tox](#) for managing test environments for Python 2.6, 2.7, 3.3, PyPy etc.
- [Pytest](#) or [Nose](#) for testing Python 2.6, 2.7, 3.3, PyPy etc.
- *Optional* support for creating a tests matrix out of dependencies and python versions.
- [Travis-CI](#) and [AppVeyor](#) for continuous testing.
- [Coveralls](#) or [Codecov](#) for coverage tracking (using [Tox](#)).
- Documentation with [Sphinx](#), ready for [ReadTheDocs](#).
- Configurations for:
 - [isort](#)
 - [bumpversion](#)
- Support for C extensions (including coverage measurement for the C code).
- Packaging and code quality checks. This template comes with a tox environment (`check`) that will:
 - Check if your `README.rst` is valid.
 - Check if the `MANIFEST.in` has any issues.
 - Run `flake8` (a combo of PEP8, pyflakes and McCabe checks)

Demmo

EuroPython 2015

Raphael Pierzina @hackebrot

HACKEBROT

[CONTACT](#) [IMPRINT](#)

[Home](#) » Archive by Category "Python"

Wrapping up Cookiedozer

February 2, 2015 By Raphael In Python

We are about to finish this tutorial on how to create a Cookiecutter template for an Android application running Kivy. If you haven't checked out [Create Your Own Cookiecutter](#) and [Extending Cookiedozer](#) yet, please have a brief look into these articles as this post draws on...

Extending Cookiedozer

January 18, 2015 By Raphael In Python

Last time we learned how to create a basic Kivy application that will be generated with Cookiecutter. The next logical steps are to add tests and docs. Additionally we will create a setup.py as well as a MAKEFILE. The template in it's current state contains the following...

Create Your Own Cookiecutter

January 7, 2015 By Raphael In Python

Are you doing the same steps over and over again every time you start a new programming project? More particularly when creating a PyPI compliant application which features a test suite along with auto-generated documentation? If so, you should definitely check out Cookiecutter....

ABOUT ME



Raphael Pierzina

I write about Python, Git, Open Source, Vim, Linux, Comic Books, Animated Films

and other things drawing my attention.



MY GITHUB PROJECTS

- [checkmate](#)
- [cookiecutter](#)
- [cookiecutter-kivy](#)
- [cookiedozer](#)
- [daumenkivy](#)
- [dw-minion](#)
- [garden.calendar](#)
- [kivy](#)
- [kivy-gallery](#)
- [kivy-gettext-example](#)

Personal Blog
www.hackebrot.de

Cookiecutter 3D Artworks by
Anselm Wagner
www.anselmwagner.com





/ hackebrot



@hackebrot

Thank you!

EuroPython 2015

Raphael Pierzina @hackebrot