

# Making Your Projects Public



John Naylor

President of RamDev

*(I'm finally doing a talk on package managers)*

# Roadmap

pip (PyPI/python)

cargo (cartes.io/rust)

npm (npmjs.com/node.js)

Nuget (nuget.org/.NET)

homebrew (macOS)

# pip

Python Packaging Authority (pypa):

- Manage Python Package Index (PyPI)
- Create and maintain tools for interacting with pypi
  - pip
  - pipenv
  - warehouse
  - twine
  - setuptools
  - Etc. etc.



# Project Structure

```
helloworld/  
|  
├── .gitignore  
├── helloworld.py  
├── LICENSE  
├── README.md  
├── requirements.txt  
├── setup.py  
└── tests.py
```

- README for description
- helloworld.py as source file
- tests.py contains project tests
- requirements.txt/Pipfile for dependencies
- **setup.py** for installation

# setup.py

```
~$ python setup.py register  
~$ python setup.py sdist  
~$ twine upload dist/*
```

```
1  
2 import pathlib  
3 from distutils.core import setup  
4  
5 # The directory containing this file  
6 HERE = pathlib.Path(__file__).parent  
7  
8 # The text of the README file  
9 README = (HERE / "README.md").read_text()  
10  
11 setup(  
12     name = 'hrpy',  
13     packages = ['hrpy'],  
14     version = '0.1.0',  
15     license='MIT',  
16     description = 'hr but written in python',  
17     long_description = README,  
18     long_description_content_type = 'text/markdown',  
19     author = 'John Naylor',  
20     author_email = 'jonaylor89@gmail.com',  
21     url = 'https://github.com/jonaylor89/hrpy',  
22     download_url = 'https://github.com/jonaylor89/hrpy/archive/v_1.0.tar.gz',  
23     keywords = ['hr', 'terminal', 'formatting'],  
24     entry_points={"console_scripts": ["hr=hrpy.hr:main"]},  
25     classifiers=[  
26         'Development Status :: 3 - Alpha',  
27  
28         'Intended Audience :: Developers',  
29         'Topic :: Software Development :: Build Tools',  
30  
31         'License :: OSI Approved :: MIT License',  
32  
33         'Programming Language :: Python :: 3',  
34         'Programming Language :: Python :: 3.4',  
35         'Programming Language :: Python :: 3.5',  
36         'Programming Language :: Python :: 3.6',  
37         'Programming Language :: Python :: 3.7',  
38     ],  
39 )  
40
```

# Creating Releases on Github

jonaylor89 / hrpy

Unwatch

1

Star

0

Fork

0

<> Code

🔔 Issues 0

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

📊 Insights

⚙️ Settings

Releases

Tags

Draft a new release

Latest release

📦 v\_0.9

🔗 0710e65

Cleaning things up

Edit

👤 jonaylor89 released this 27 days ago · 1 commit to master since this release

v\_0.9

style

▼ Assets 2

📄 Source code (zip)

📄 Source code (tar.gz)

📦 v\_0.8

🔗 c703ae4

Fix Path

Edit

👤 jonaylor89 released this on Dec 8, 2018 · 9 commits to master since this release

# PyPI and Installation

[Help](#)[Donate](#)[Log in](#)[Register](#)

## hrpy 0.0.9

`pip install hrpy`[Latest version](#)

Last released: Mar 27, 2019

hr but written in python

```
CMSC508 on  master  
→ pip install hrpy
```

# cargo

“Cargo is the Rust *package manager*. Cargo downloads your Rust package’s dependencies, compiles your packages, makes distributable packages, and uploads them to crates.io, the Rust community’s *package registry*.”

- [doc.rust-lang.org/cargo/](https://doc.rust-lang.org/cargo/)





# Basic Cargo

Cargo is a suite for rust programming

**crates.io**  
Rust Package Registry

Click or press 'S' to search...

[Browse All Crates](#)[Docs](#)[Log in with GitHub](#)

# The Rust community's crate registry

[Install Cargo](#)[Getting Started](#)

Instantly publish your crates and install them. Use the API to interact and find out more information about available crates. Become a contributor and enhance the site with your work.

**982,464,054** Downloads**25,083** Crates in stock

## New Crates

nldm (0.1.0)



oxyengine-network-backend-we...



combine-proc-macro (0.2.0)



## Most Downloaded

libc (0.2.53)



rand (0.6.5)



bitflags (1.0.4)



## Just Updated

print\_perf (0.1.9)



oxyengine (0.3.3)



libui-sys (0.1.3)



# Packaging

“This command will create a distributable, compressed .crate file with the source code of the package in the current directory. The resulting file will be stored in the target/package directory.”

- [doc.rust-lang.org/cargo](https://doc.rust-lang.org/cargo)

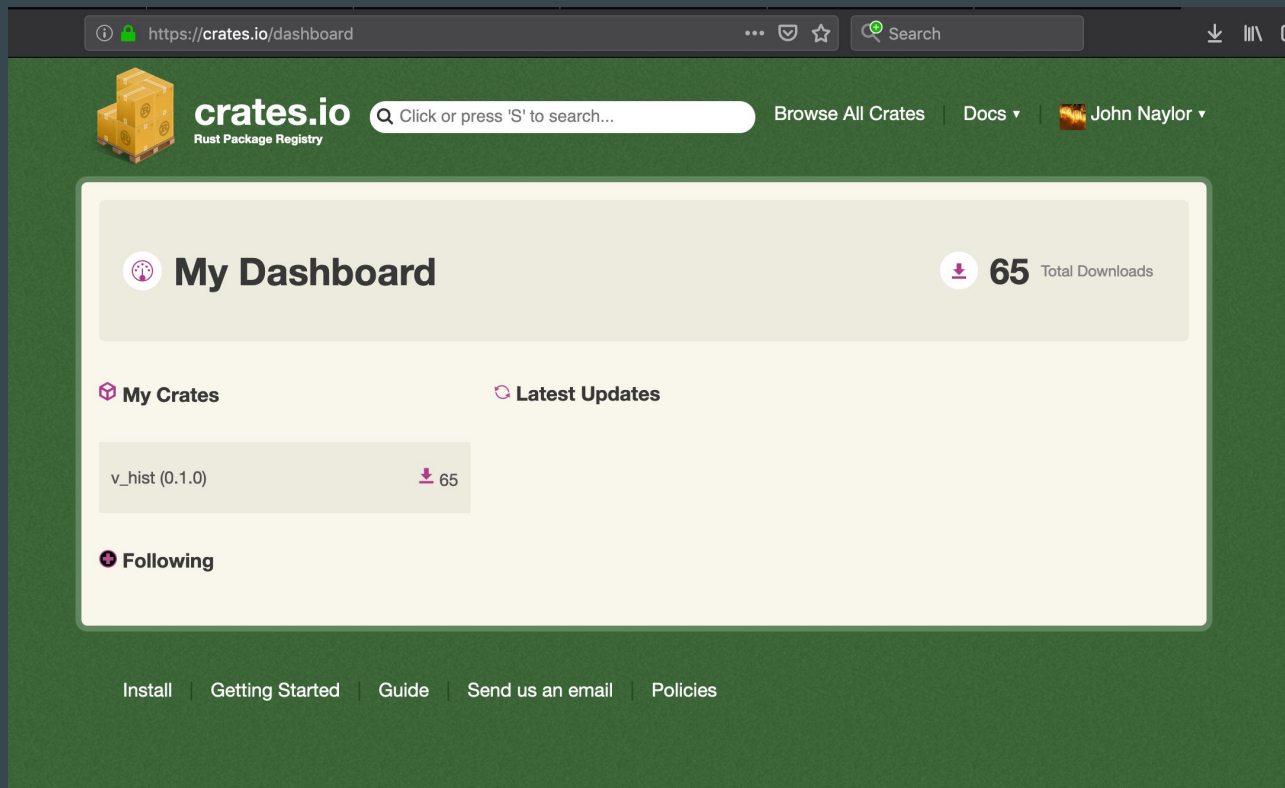
# Publishing and Yanking

cargo publish

- Upload Package

cargo yank

- Remove Package



npm



“Serving 500 of the Fortune 500 (and everyone else)” - [npmjs.com](https://npmjs.com)

**npmjs.org is a meme**

# is-odd

npm v3.0.1

downloads 4M/month

downloads 126M

Travis passing

Returns true if the given number is odd, and is an integer that does not exceed the JavaScript MAXIMUM\_SAFE\_INTEGER.

Please consider following this project's author, [Jon Schlinkert](#), and consider starring the project to show your ❤️ and support.

## Install

Install with **npm**:

```
$ npm install --save is-odd
```

## Usage

Works with strings or numbers.

```
const isOdd = require('is-odd');

console.log(isOdd('1')); //=> true
console.log(isOdd('3')); //=> true

console.log(isOdd(0)); //=> false
console.log(isOdd(2)); //=> false
```

install

```
> npm i is-odd
```

↓ weekly downloads

840,656



version

3.0.1

license

MIT

open issues

0

pull requests

5

homepage

[github.com](#)

repository

 [github](#)

last publish

a year ago

collaborators



Test with RunKit

[Report a vulnerability](#)

# is-even

npm v1.0.0

downloads 220k/month

downloads 2M

Travis passing

Return true if the given number is even.

## Install

Install with **npm**:

```
$ npm install --save is-even
```

## Usage

```
var isEven = require('is-even');

isEven(0);
//=> true
isEven('1');
//=> false
isEven(2);
//=> true
isEven('3');
//=> false
```

## About

install

```
> npm i is-even
```

± weekly downloads

44,460



version

1.0.0

license

MIT

open issues

0

pull requests

1

homepage

github.com

repository

 github

last publish

2 years ago

collaborators



Test with RunKit

Report a vulnerability



# is-even

1.0.0 • Public • Published 2 years ago

Readme

1 Dependencies

12 Dependents

4 Versions

## Dependencies (1)

is-odd

## Dev Dependencies (2)

gulp-format-md mocha

install

```
> npm i is-even
```

± weekly downloads

44,460



version

1.0.0

license

MIT

open issues

0

pull requests

1

homepage

github.com

repository

 github

last publish

2 years ago

npm's 2019 JavaScript ecosystem survey analysis is now available! [Get your copy here »](#)



## memes

0.1.0 • [Public](#) • Published 3 years ago

[Readme](#)[4 Dependencies](#)[0 Dependents](#)[1 Versions](#)

# The Meme Machine

*Welcome to the next era of the internet. This software will change your life. A new dawn of dank memes awaits us. Are you ready?*

For years we have relied on proprietary software like Photoshop, Flash, and Premiere to create stupid internet videos. This project puts an end to all that. No longer do we have to let  big software companies  control the free exchange of memes on our internet.

## Installation

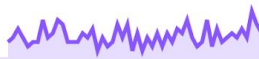
You can use this software from the comfort of your browser without installing anything by visiting <https://majorleague.website>

install

```
> npm i memes
```

↓ weekly downloads

4



version

0.1.0

license

none

last publish

3 years ago

collaborators



# trolld

A package that randomly "throws" fake errors that look very scary. If used discreetly it could make for a very fun afternoon of watching your co-worker panic!

## How to use

Simple require the package as a function (preferably discreetly somehow):

```
require('trolld')([chance])
```

 - Replace [chance] with a 0-100 chance to trigger on load. If not provided, one will be provided.

### Keywords

troll fun errors coworker funny

install

```
> npm i trolld
```

↓ weekly downloads

10



version

1.0.2

license

MIT

open issues

0

pull requests

0

homepage

github.com

repository

 github

last publish

a year ago

collaborators



# Publishing

## Unscoped

- Publishing one stand alone package
- For npm public registry

Ex.

- npm install yourPackage

## Scoped

- Grouping a set of packages together for publishing
- For npm Enterprise

Ex.

- npm install @yourScope

# Homebrew



## Homebrew terminology

Term	Description	Example
Formula	The package definition	<code>/usr/local/Homebrew/Library/Taps/homebrew/homebrew-core/Formula/foo.rb</code>
Keg	The installation prefix of a <b>Formula</b>	<code>/usr/local/Cellar/foo/0.1</code>
opt prefix	A symlink to the active version of a <b>Keg</b>	<code>/usr/local/opt/foo</code>
Cellar	All <b>Kegs</b> are installed here	<code>/usr/local/Cellar</code>
Tap	A Git repository of <b>Formulae</b> and/or commands	<code>/usr/local/Homebrew/Library/Taps/homebrew/homebrew-core</code>
Bottle	Pre-built <b>Keg</b> used instead of building from source	<code>qt-4.8.4.mavericks.bottle.tar.gz</code>
Cask	An <a href="#">extension of Homebrew</a> to install macOS native apps	<code>/Applications/MacDown.app/Contents/SharedSupport/bin/macdown</code>
Brew Bundle	An <a href="#">extension of Homebrew</a> to describe dependencies	<code>brew 'myservice', restart_service: true</code>

# Publishing on homebrew-core

## Grab the URL

Run `brew create` with a URL to the source tarball:

```
brew create https://example.com/foo-0.1.tar.gz
```

This creates `/usr/local/Homebrew/Library/Taps/homebrew/homebrew-core/Formula/foo.rb` and opens it in your `EDITOR`. It'll look something like:

```
class Foo < Formula
  desc ""
  homepage ""
  url "https://example.com/foo-0.1.tar.gz"
  sha256 "85cc828a96735bdafcf29eb6291ca91bac846579bcef7308536e0c875d6c81d7"

  # depends_on "cmake" => :build

  def install
    # ENV.deparallelize
    system "./configure", "--disable-debug",
                        "--disable-dependency-tracking",
                        "--disable-silent-rules",
                        "--prefix=#{prefix}"

    # system "cmake", ".", *std_cmake_args
    system "make", "install"
  end

  test do
    system "false"
  end
end
```

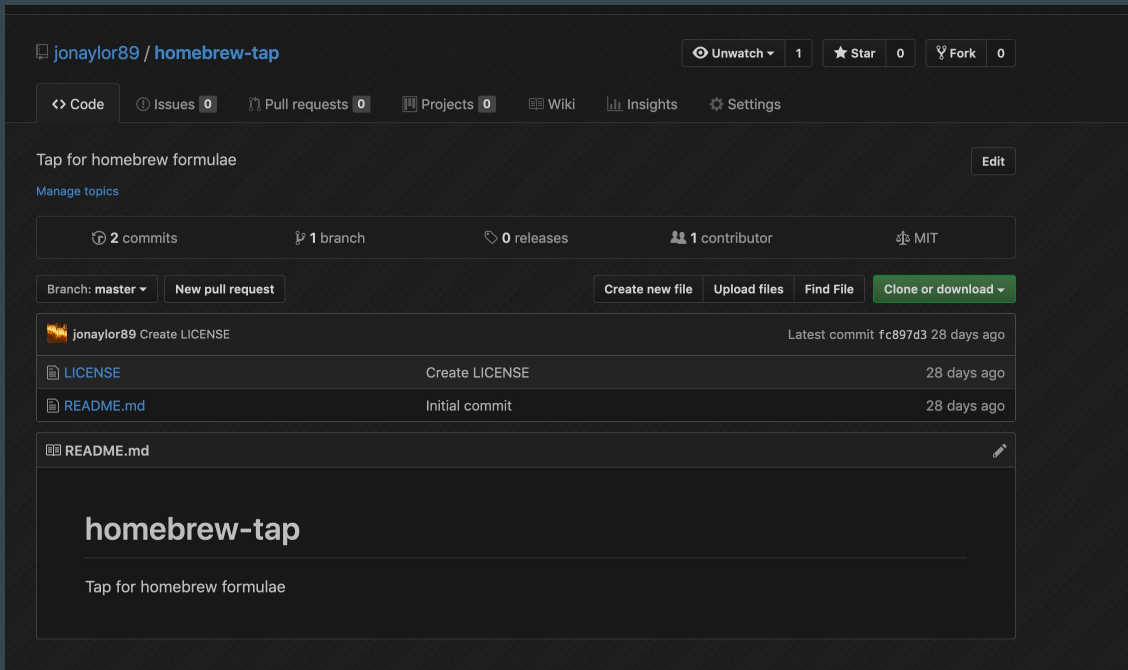
Publishing to homebrew is equivalent to a file to committing to github!

1. `~$ brew create <your-url>`
2. Pull homebrew-core
3. Push your formula in a separate branch

# Setting up a tap

Taps are as simple as making a github repository for your formulae

~\$ brew tap jonaylor89/tap



# Tapping and Installing

Download from other users taps

```
$ brew tap  
homebrew/core  
mistydemeo/tigerbrew  
dunn/emacs
```



# Other Popular Package Managers

gem (Ruby)

apt (debian)

composer (php)

yum (RedHat)

snap (Ubuntu)

chocolatey (Windows)

pacman/packer/yay (Arch/AUR)

go get (golang)

NuGet (.NET)

# Questions?