

# Michael Foord

<https://agileabstractions.com/>



- Python trainer & contractor
- Python developer since 2002
- Core Python Developer
- Author of IronPython in Action
- Creator of “unittest.mock”
- Twitter: @voidspace

finite\_difference.rsl\* - Resolver One (Evaluation)

File Edit View Format Data Worksheet Help

Find:

=[val for val in <Model>.Cols['Plot Result'] if val > A1]

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1																									
2																									
3	Main Menu																								
4	Modelin																								
5	Load Model																								
6	Save Model																								
7																									
8	Calculations																								
9	Solve Model																								
10																									
11	Visualization																								
12	Plot Solution (3D)																								
13	Plot Solution (2D)																								
14	Save 3D Graph as PNG																								
15	Save 2D Graph as PNG																								
16	Help																								

Resolver One Console for finite\_difference.rsl

```

'_module_', '_new_', '_reduce_', '_reduce_ex_', '_repr_', '_setattr_',
'_str_', '_weakref_']
>>> def f(sheet):
...     return max(val for val in sheet if isinstance(val, (int, float)))
...
>>> f(workbook['Solution Data'])
10

help(worksheet)

```

Model Solution Data Solution Data Plot (RESULT)

```

'GetType', 'MemberwiseClone', 'Populate', 'ReferenceEquals', 'SelectedWorksheet', 'ToString', 'UpdateTraitDefaults', '__class__', '__delattr__',
'dict__', 'doc', 'getattribute', 'getitem', 'hash', 'init', 'iter', 'module', 'new', 'reduce', 'reduce_ex',
'repr', 'setattr', 'str', 'weakref_']
10

```

# License (not editable)...

# Import statements (not editable)...

# Worksheet creation (not editable)

```

workbook.AddWorksheet("Model")
workbook["Model"].Bounds = (1, 3, 30, 23)
workbook["Model"].HeaderRow = workbook["Model"].RowsBy

workbook.AddWorksheet("Solution Data")

```

# Pre-constants user code

```

sheet = workbook['Model']
import clr
clr.AddReference('System.Windows.Forms')
from System.Windows.Forms import Application, Form, Pr
class ProgressDialog(Form):

```

Results...

Solution Plot

X axis

Y axis

Z axis

*Agile, DevOps and Scrum, along with the tooling and technologies of modern engineering enable us to tackle the difficult problems of building and running robust systems with confidence. – Michael Foord*

# Some Core Values from Agile

- Testing: quality and confidence
- MVP, do the simplest thing, grow organically and refactor
- Tooling: version control, Kanban, issue tracking
- Iterations: able to change direction (agile)
- Estimation and prioritisation

# Kanban with Jira

Projects / Sphinx / OTS

## Sphinx Sprint 31

Support FSL where needed and continue work on PS into Vue3

🕒 3 days remaining



M

DJ

EP

N

R

Epic ▾

Type ▾

More ▾

GROUP BY

Stories ▾

TO DO 5

Create lead in Salesforce from Gigago...  
GIGAGO  
🚩 ●●●● ▾  
📌 OTS-512

Add OTS trigger pipeline job t...  
2 ●●●● ▾  
📌 OTS-579

IN PROGRESS 4

Implement an email solution to raise critic...  
OTS - GENERAL  
🚩 8 ●●●● ▾  
📌 OTS-499 M

Create a new repository for the postal m...  
DIRECT POSTAL MAIL  
8 ●●●● ▾

PEER REVIEW

AWAITING DEPL... 1

Add parameter that trims...  
OBJ PERFORMANCE  
3 ●●●● ▾  
📌 OTS-559 M

QA 1

Add automated tests on the...  
8 ●●●● =  
📌 OTS-546 EP

DONE 7

Capture new Accounts added to...  
NETADMIN 9.1 NET...  
5 ●●●● ▾  
📌 OTS-493 EP

Git hook for automation pipelines  
QA AUTOMATION  
▾

Python Language Website x +

https://web.archive.org/web/20020124171739/http://python.org/

Apps Subscribe Note View RSS GReader gmail Blog reddit BBC NEWS Flickr Planet Py Other bookmarks

**Python**

Home Help Search News Download Community Documentation SIGs

**Special topics**

- [Beginners Guide](#)
- [Topic Guides](#)
- [Python 2.2](#)
- [Python 2.1.2](#)
- [Python 2.0.1](#)
- [Python 1.6.1](#)
- [Python 1.5.2](#)
- [MacPython](#)
- [Parrot](#) (Python and Perl)
- [Jython](#) (Python in Java)
- [Pipppy](#) (Python for Palm)
- [Tkinter](#) (Tcl/Tk)
- [Editing Python code](#)
- [IDLE](#)
- [IDLE fork](#)
- [Contributed software](#)
- [Python at SourceForge](#)
- [PEPs](#)

**Popular links**

- [Python Cookbook](#)
- [FAQ wizard](#)
- [Conferences](#)
- [- Python10](#)
- [Python Job Board](#)

**Python**

www.python.org website hosted by [XS4ALL](#)

**ANNOUNCEMENTS**

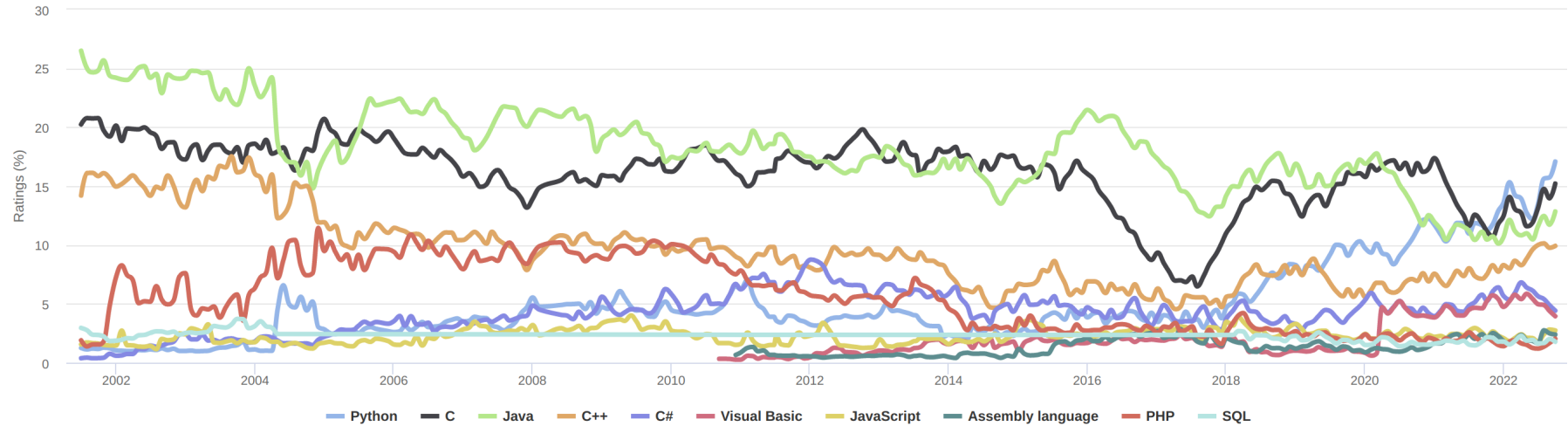
- The [O'Reilly Open Source Conference](#) is requesting proposals for tutorials, presentations and panels.
- [Python 2.1.2 is out](#) - a bugfix release for Python 2.1.
- [Jython 2.1 is out](#) - the final release of Jython 2.1 (compatible with Python 2.1(.2)).
- [Python 2.2 is out](#) - the final release of Python 2.2.
- [New Python gear](#) - HackerThreads has T-shirts and hats.
- [Registration is open](#) for the [Python10 Conference](#). Keynote speakers are Tim Berners-Lee and Andrew Koenig!
- Check out our new [Beginners Guide](#) - all information for budding programmers collected together on one page.
- Check out the new [IDLE fork](#) website!
- [Python 2.1.1 is out](#) - a bugfix release for Python 2.1 (final).

Welcome to the official website for the Python language. The following items are accessible from the top of each page on this site:

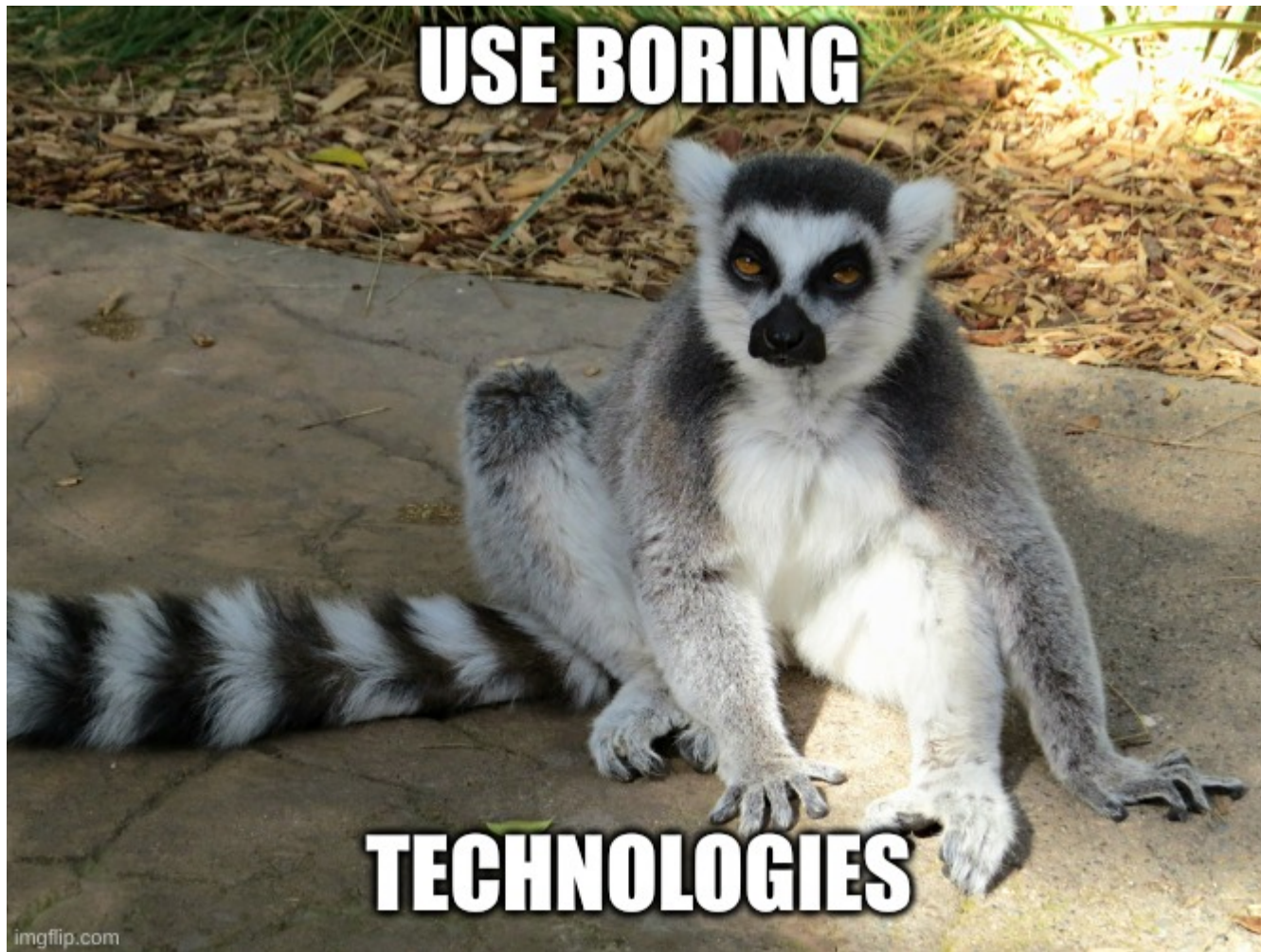
- [Home](#) - this page: orientation, announcements, news
- [Tutorial](#) - learn Python in an afternoon!
- [Help](#) - if you're new here, or simply lost

## TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)





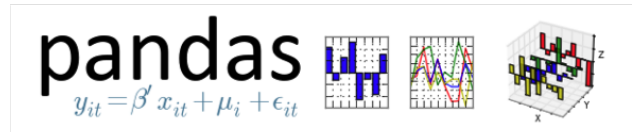




# Python is Boring Technology



IP[y]: IPython  
Interactive Computing





git



# Version Control

- A backup
- Always know which is the latest version
- Be able to go back to earlier versions
- Merge changes from several sources

# PYTHON BLACK

The Uncompromising  
Code Formatter



## About Python Package Metadata

Feb 14, 2020

Since this topic keeps coming up, I'd like to briefly share my thoughts on Python package metadata because it's – as always – more complex than it seems.

When I say metadata I mean mostly the version so I will talk about it interchangeably. But the description, the license, or the project URL are also part of the game.

The overarching problem is that we have two places where we may need that metadata:

1. In the packaging mechanism. *setuptools*, *pip*, *flit*, *poetry*, et al

## A quick-and-dirty guide on how to install packages for Python



BRETT CANNON  
21 JAN 2020 • 5 MIN READ

When people start learning Python, they often will come across a package they want to try and it will usually start with "just `pip install` it!" The problem with that advice is it's a very simplistic view of how to manage packages and can actually lead to problems down the road. And while there is a [tutorial on installing packages at packaging.python.org](https://packaging.python.org), it might be a bit intimidating for some if they are just looking to quickly get up and going.

If you just want to start poking at Python and want to avoid the pitfalls to installing packages globally, it only takes 3 steps to do the right thing.

### Summary

1. Create a [virtual environment](#), e.g. `python3.8 -m venv .venv` (substitute `py -3.8` for `python3.8` if necessary)
2. Activate the virtual environment, e.g. `source .venv/bin/activate.fish` (assuming you are using the [fish shell](#))
3. Install the packages you want, e.g. `python -m pip install --upgrade pip`

# Dependency Management

```
git clone https://github.com/voidspace/...  
cd repository  
python -m venv .env  
.env\Scripts\activate  
pip install -r requirements.txt
```

cloud

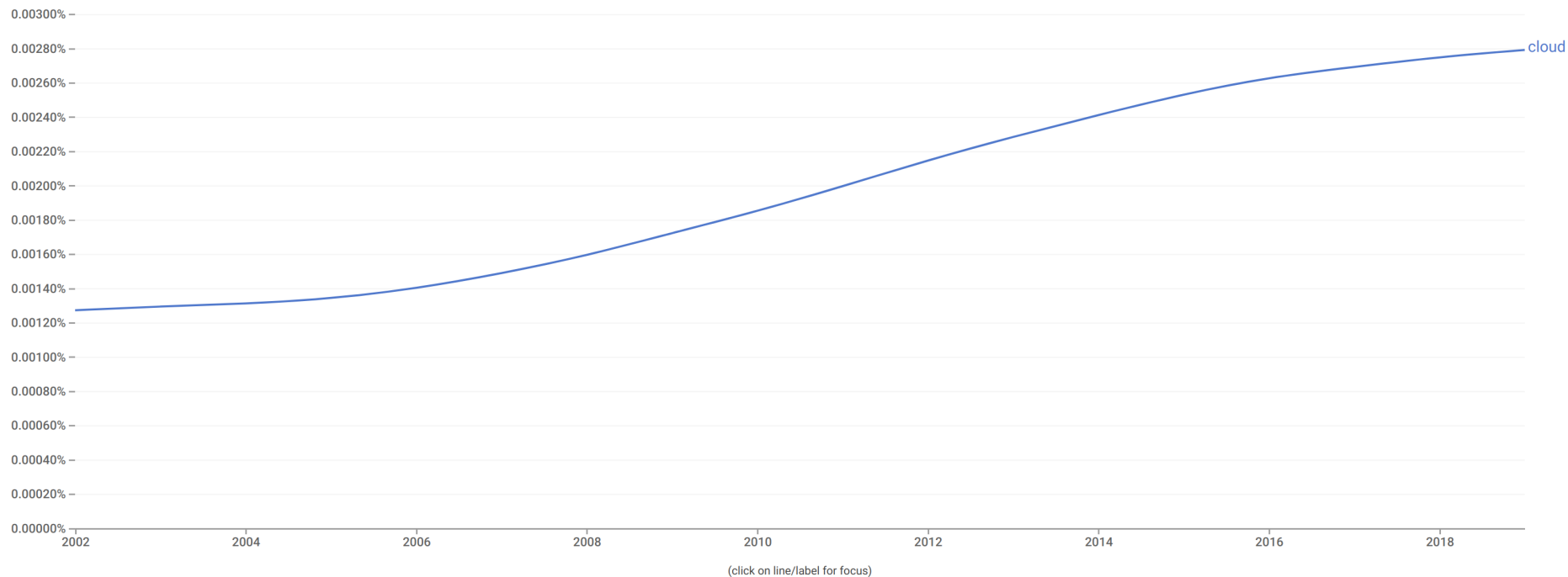


2002 - 2019

English (2019)

Case-Insensitive

Smoothing





● devops  
Search term

+ Compare

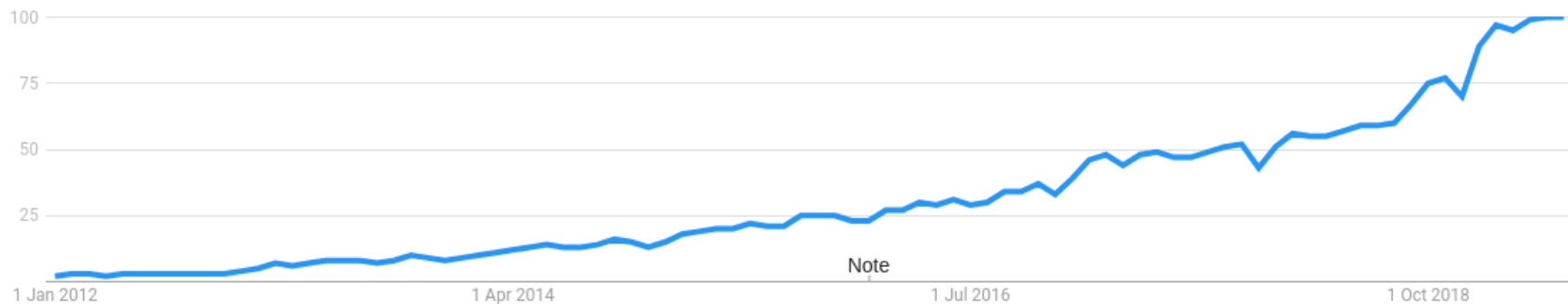
Worldwide ▼

01/01/2012 - 21/06/2019 ▼

All categories ▼

Web Search ▼

Interest over time ?



python ai  
Search term



+ Compare

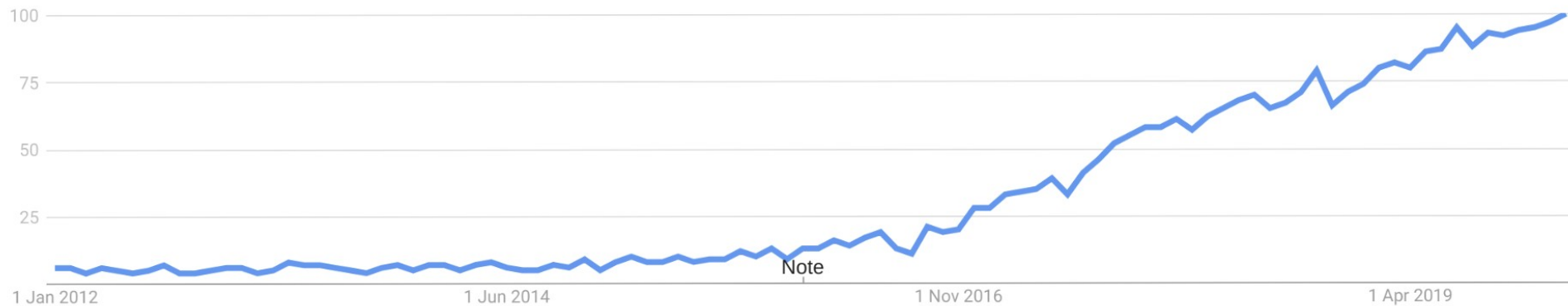
Worldwide ▼

01/01/2012 - 20/02/2020 ▼

All categories ▼

Web Search ▼

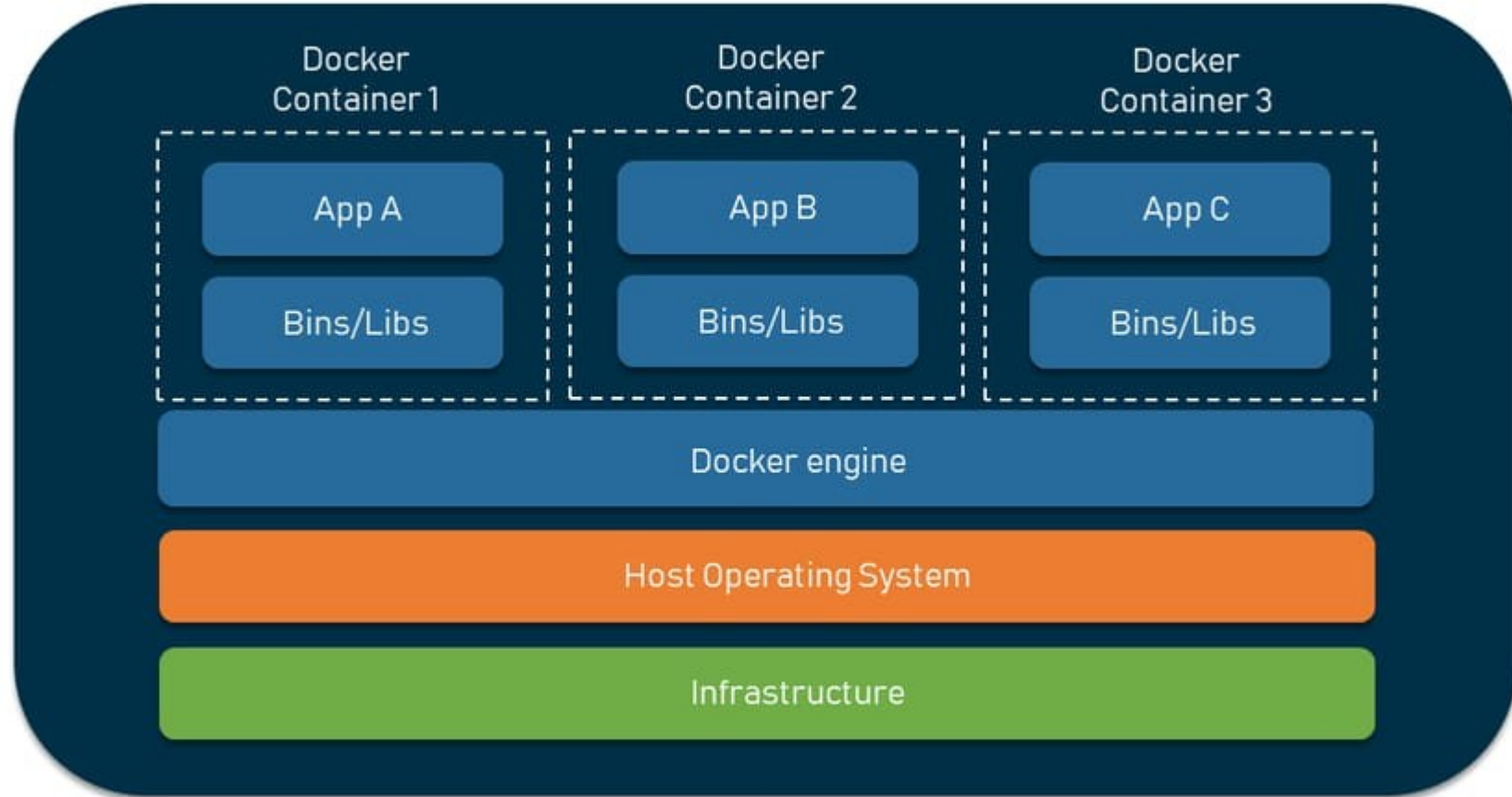
Interest over time ?



# The Cloud and DevOps

- The cloud enabled devops
- Containers as the unit of deployment
- Repeatable automated builds and deployment
- Dev build artefact similar to prod environment
- Docker and Docker images! (and then kubernetes)

# DOCKER CONTAINERS



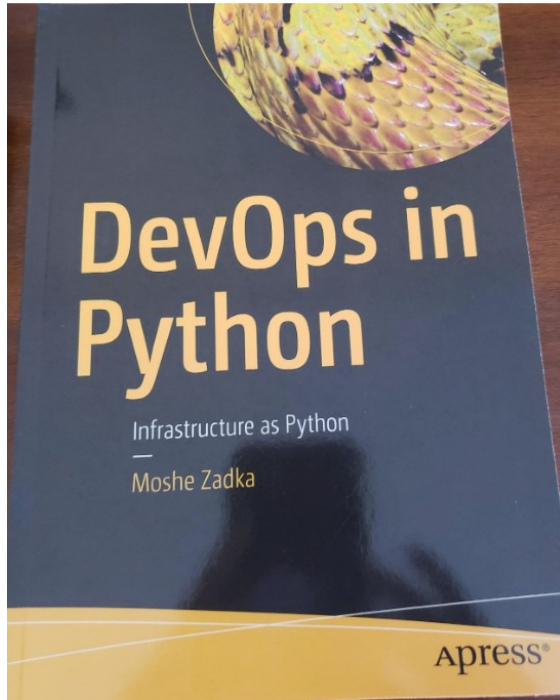
# DevOps

- Running services as well as building them
- Automated workflows
  - Automated testing: deploy with confidence
  - Push button deploy and rollback
  - Continuous Integration/Continuous Deployment
- Caring for the system for the whole lifecycle
- Tooling like docker, kubernetes, ansible, infrastructure as code



**Moshe Zadka** • 1st  
Senior Site Reliability Engineer at SurveyMonkey  
1w

My new book! [#python](#)



© 2019

## DevOps in Python

Infrastructure as Python

Authors: Zadka, Moshe

[Download source code](#)

[Free Preview](#)

- ✓ Builds on readers basic familiarity with Python to focus on those parts essential to DevOps

[see more benefits](#)

### About this book

### About the authors

Explore and apply best practices for efficient application deployment. This book draws upon author Moshe Zadka's years of Dev Ops experience and focuses on the parts of Python, and the Python ecosystem, that are relevant for DevOps engineers.

You'll start by writing command-line scripts and automating simple DevOps-style tasks. You'll then move on to more advanced cases, like using Jupyter as an auditable remote-control panel, and

[Show all](#)

[Table of contents](#) (13 chapters)

## Disposable virtual environments



## Python for DevOps

Learn Ruthlessly Effective Automation

By [Grig Gheorghiu](#), [Noah Gift](#), [Kennedy Behrman](#), [Alfredo Deza](#)

**Publisher:** [O'Reilly Media](#)

**Release Date:** December 2019

**Pages:** 506

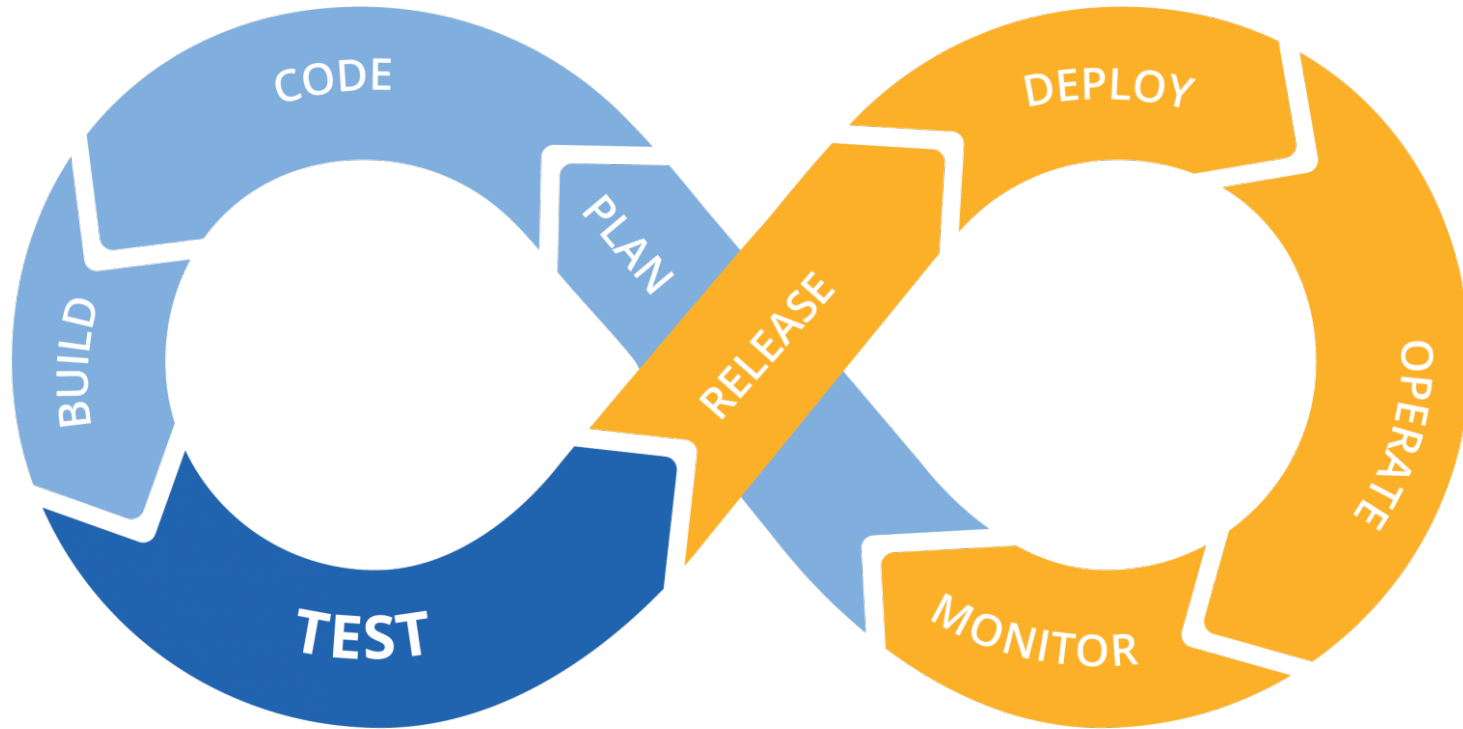
Much has changed in technology over the past decade. Data is hot, the cloud is ubiquitous, and many organizations need some form of automation. Throughout these transformations, Python has become one of the most popular languages in the world. This practical resource shows you how to use Python for everyday Linux systems administration tasks with today's most useful DevOps tools, including Docker, Kubernetes, and Terraform.

Learning how to interact and automate with Linux is essential for millions of professionals. Python makes it much easier. With this book, you'll learn how to develop software and solve problems using containers, as well as how to monitor, instrument, load-test, and operationalize your software. Looking for effective ways to "get stuff done" in Python? This is your guide.

## Great interviews with prominent Python developers



# CI/CD



Government Digital Services, Singapore

## Dictionary



# process engineering

*noun*

the branch of engineering that is concerned with industrial processes, especially continuous ones such as the production of petrochemicals.



Translations, word origin and more definitions

*The deliberate design of processes  
and workflows, supported by  
tooling, to enable the continuous  
and sustained delivery and  
maintenance of a quality system.*

*The modern Python ecosystem and infrastructure, along with conventions and understanding from the wider programming community, means that the “how” of software engineering is often a solved problem. What and why still remain to be solved...*

*Agile, DevOps and Scrum, along with the tooling and technologies of modern engineering enable us to tackle the difficult problems of building and running robust systems with confidence –  
Michael Foord*

`sys.exit(0)`