Why mastering Python is impossible, and why that's OK

by Rodrigo Girão Serrão

UCL Data and Insight CoP

About me

Rodrigo Girão Serrão

Formal education: maths

Writing Python since 2012

Training/teaching:

- APL (Dyalog Ltd.)
- Python, maths, etc (mathspp.com)

@mathsppblog

“master”, verb

“master”, verb

“master”, verb — to learn or understand something completely

“master”, verb

“master”, verb — to learn or understand something completely

“master”, verb

- Python is an evolving language
  - (~1 release/year)

"master", v

- Python is
  - (~1 relea

What's New in Python

The "What's New in Python" series of essays takes tours through the most important changes between major Python versions. They are a "must read" for anyone wishing to stay up-to-date after a new release.

“master”, verb

- Python is an evolving language
  - (~1 release/year)
- Python is just *too big*

Python is big

Python is big

Built-in functions?

71

*Python 3.10, as of April 2022

Pytho

Built-i

71

Built-in Functions

The Python interpreter has a number of functions and types built into it that are always available. They are listed here in alphabetical order.

Built-in Functions

A
abs()
aiter()
all()
any()
anext()
ascii()

B
bin()
bool()
breakpoint()
bytearray()
bytes()

C
callable()
chr()
classmethod()
compile()
complex()

D
delattr()
dict()
dir()
divmod()

E
enumerate()
eval()
exec()

F
filter()
float()
format()
frozenset()

G
getattr()
globals()

H
hasattr()
hash()
help()
hex()

I
id()
input()
int()
isinstance()
issubclass()
iter()

L
len()
list()
locals()

M
map()
max()
memoryview()
min()

N
next()

O
object()
oct()
open()
ord()

P
pow()
print()
property()

R
range()
repr()
reversed()
round()

S
set()
setattr()
slice()
sorted()
staticmethod()
str()
sum()
super()

T
tuple()
type()

V
vars()

Z
zip()

_
__import__()

*Python 3.10, as of April 2022

Python is big

Built-in types?

- *int* / *float* / *complex*
- *list* / *tuple*
- *dict*
- *set* / *frozenset*
- *str*

*Python 3.10, as of April 2022

Python is big

Built-in types + methods?

- *int* / *float* / *complex*
- *list* / *tuple*
- *dict*
- *set* / *frozenset*
- *str*

*Python 3.10, as of April 2022

Python is big

Built-in types + methods?

- *int* (10) / *float* (7) / *complex* (3)
- *list* / *tuple*
- *dict*
- *set* / *frozenset*
- *str*

*Python 3.10, as of April 2022

Python is big

Built-in types + methods?

- *int* (10) / *float* (7) / *complex* (3)
- *list* (11) / *tuple* (2)
- *dict*
- *set* / *frozenset*
- *str*

*Python 3.10, as of April 2022

Python is big

Built-in types + methods?

- *int* (10) / *float* (7) / *complex* (3)
- *list* (11) / *tuple* (2)
- *dict* (11)
- *set* / *frozenset*
- *str*

*Python 3.10, as of April 2022

Python is big

Built-in types + methods?

- *int* (10) / *float* (7) / *complex* (3)
- *list* (11) / *tuple* (2)
- *dict* (11)
- *set* (17) / *frozenset* (8)
- *str*

*Python 3.10, as of April 2022

Python is big

Built-in types + methods?

- *str* (47!)
  - capitalize, casefold, center, count, encode, endswith, expandtabs, find, format, format_map, index, isalnum, isalpha, isascii, isdecimal, isdigit, isidentifier, islower, isnumeric, isprintable, isspace, istitle, isupper, join, ljust, lower, lstrip, maketrans, partition, removeprefix, removesuffix, replace, rfind, rindex, rjust, rpartition, rsplit, rstrip, split, splitlines, startswith, strip, swapcase, title, translate, upper, zfill

*Python 3.10, as of April 2022

Python is big

Built-in types + methods?

- *int* (10) / *float* (7) / *complex* (3)
- *list* (11) / *tuple* (2)
- *dict* (11)
- *set* (17) / *frozenset* (8)
- *str* (47)

*Python 3.10, as of April 2022

Python is big

Standard library?

*Python 3.10, as of April 2022

Pyth

Rodrigo 🐍📝
@mathsppblog

Did you know that Python 🐍 has over 240 modules in the standard library?

Here is a MEGA thread 🧵 with a super high level overview of those modules...

... but before that, I challenge you to name as many modules by heart as possible! 💬👇

Let's see who gets more!

6:02 AM · Aug 24, 2021 · Twitter Web App

269 Retweets **24** Quote Tweets **1,013** Likes

*Python 3.10, as of April 2022

Pyth

Rodrigo
@mathsppblog

👉 Superseded Modules
🔗 docs.python.org/3/library/supe...

These modules are here for backward compatibility.

2 4 8 optparse — Parser for command line options
👆 Use `argparse` instead.

2 4 9 imp — Access the import internals
👆 Use `importlib` instead.

7:51 AM · Aug 24, 2021 · Twitter Web App

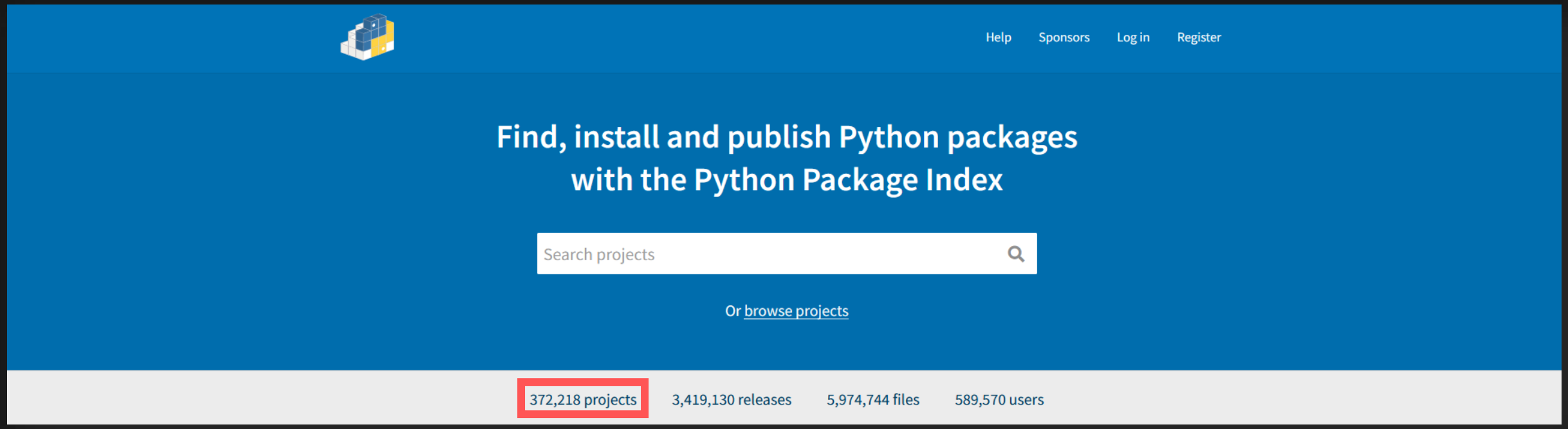
*Python 3.10, as of April 2022

Python is big

Other packages?

On pypi.org: 372,218 and counting.

Python is big



Python is big

> WHAT WE KNOW IS A DROP, WHAT WE DON'T KNOW IS AN OCEAN.

— Sir Isaac Newton (?)

Python is big

There is a lot to learn:

- don't hoard knowledge;
- learn the tools you *need* & *write code*;
- keep improving *gradually* and *consistently*.

Improving gradually & consistently

Improving gradually & consistently

Programmer's goal: use the right tool for the job.
Aka, “don't reinvent the wheel”.

Improving gradually & consistently

Main objective: getting exposure

- functions;
- methods;
- features;
- quirks;
- modules;
- ...

Improving gradually & consistently

Main objective: getting exposure

Main benefit: increased awareness

- functions;
- methods;
- features;
- quirks;
- modules;
- ...

Disclaimer:

Unstructured tips ahead.

Mileage may vary.

Documentation

Documentation

Your BFF: docs.python.org

Documentation

docs.python.org personal fav pages:

- built-in functions;
- built-in types;
- module index; and
- data model.

Download

Download these documents

Docs by version

Python 3.11 (in development)
Python 3.10 (stable)
Python 3.9 (stable)
Python 3.8 (security-fixes)
Python 3.7 (security-fixes)
Python 3.6 (EOL)
Python 3.5 (EOL)
Python 2.7 (EOL)
All versions

Other resources

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

Python 3.10.4 documentation

Welcome! This is the official documentation for Python 3.10.4.

Parts of the documentation:

Experiment

Experiment

1. Use the REPL.
2. Use the REPL.
3. Use the REPL.

Experiment

Inside the REPL:

1. use `help`;
2. use `dir`; and
3. use `rich`.

```
>>> from rich import pretty, traceback
>>> pretty.install(); traceback.install();
<built-in function excepthook>
>>> {42: True, "oi": None}
{42: True, 'oi': None}
>>> 1/0
╭──────────────────────────── Traceback (most recent call las
│ <stdin>:1 in <module>
╰──────────────────────────────────────────────────────────
ZeroDivisionError: division by zero
>>> help(dir)
Help on built-in function dir in module builtins:

dir( ... )
    dir([object]) → list of strings
```

Teach

Teach

Insert Richard Feynman quote here.

Teach

Where to teach:

- Online (directly):
  - Stack Overflow [python]: https://stackoverflow.com/
  - Reddit r/Python: https://www.reddit.com/r/python
  - Reddit r/learnpython: https://www.reddit.com/r/learnpython/
  - ...
- Online (indirectly):
  - Blog / website / ...
- In person:
  - Meetups / classes / programming buddies / ...

Read code

Read code

Code is read more often than it is written.

Read code

Intentionally read code from others:

- packages you use;
- the standard library;
- …

Search or jump to... / Pull requests Issues Marketplace Explore

python / cpython Public

Code | Issues 5k+ | Pull requests 1.5k | Actions | Projects 2 | Se

main | cpython / Lib /

AbhigyanBose gh-91832: Add 'required' attr to ar

..

- __phello__
- asyncio
- collections
- concurrent
- ctypes
- curses
- dbm
- distutils
- email
- encodings
- ensurepip
- html
- http
- idlelib
- importlib
- json
- lib2to3

Lib

New | Sort | View

This PC › Windows-SSD (C:) › Program Files › Python310 › Lib ›

Quick access
- Desktop
- Downloads
- Documents
- Pictures
- Videos
- mathspp
- rodri
- tmp
- Dyalog
- Programming
- 20220400_workshop_simcorp
- 20220429_ucl_cop_why_mast
- resources
- thumbnails

OneDrive - Personal

This PC

Network

Linux

Name	Date modified
__pycache__	21-Oct-21 07:42
asyncio	21-Oct-21 07:41
collections	21-Oct-21 07:41
concurrent	21-Oct-21 07:41
ctypes	21-Oct-21 07:41
curses	21-Oct-21 07:41
dbm	21-Oct-21 07:41
distutils	21-Oct-21 07:41
email	21-Oct-21 07:41
encodings	21-Oct-21 07:41
ensurepip	21-Oct-21 07:41
html	21-Oct-21 07:41
http	21-Oct-21 07:41
idlelib	21-Oct-21 07:42
importlib	21-Oct-21 07:41
json	21-Oct-21 07:41
lib2to3	21-Oct-21 07:41
logging	21-Oct-21 07:41
msilib	21-Oct-21 07:41
multiprocessing	21-Oct-21 07:41
pydoc_data	21-Oct-21 07:41
site-packages	21-Oct-21 07:42
sqlite3	21-Oct-21 07:41
test	21-Oct-21 07:41
tkinter	21-Oct-21 07:42
turtledemo	21-Oct-21 07:42
unittest	21-Oct-21 07:41
urllib	21-Oct-21 07:41
venv	21-Oct-21 07:41

202 items |

Standard Library

Standard Library

The standard library is a gold mine.

Some (personal) favs:

- `collections`
- `functools`
- `itertools`
- `random`
- `pathlib`
- `...`

Other libraries & PyPI

Other libraries & PyPI

300k+ packages to play with.

Big names include:

- Django / Flask / FastAPI (web)
- NumPy / Pandas (crunch numbers/data)
- Tensorflow / PyTorch (machine learning)
- Pygame / Arcade (games)
- ...

Depends on *your personal goals*.

Learn other languages (?!)

Learn other languages (?!)

> A LANGUAGE THAT DOESN'T AFFECT THE WAY YOU THINK ABOUT PROGRAMMING, IS NOT WORTH KNOWING.
>
> — Alan J. Perlis

Learn other languages (?!)

The greatest impact:

- APL
- Haskell

Recap

- You can't really master Python
  - Always evolving
  - Already too big
- You can keep learning
  - Read the docs
  - Experiment (use the REPL!)
  - Teach others
  - Read code
  - Learn new modules
  - Learn other languages (?!)

References

- Why mastering Python is impossible, and why that's ok, https://mathspp.com/blog/pydonts/why-mastering-python-is-impossible
- Python is a big language, https://mathspp.com/blog/twitter-threads/python-is-a-big-language
- Why APL is a language worth knowing, https://mathspp.com/blog/why-apl-is-a-language-worth-knowing
- Boost your productivity with the REPL, https://mathspp.com/blog/pydonts/boost-your-productivity-with-the-repl

pydonts.com

@mathsppblog

mathspp.com/subscribe

/mathspp/talks

email

rodrigo@mathspp.com

name site