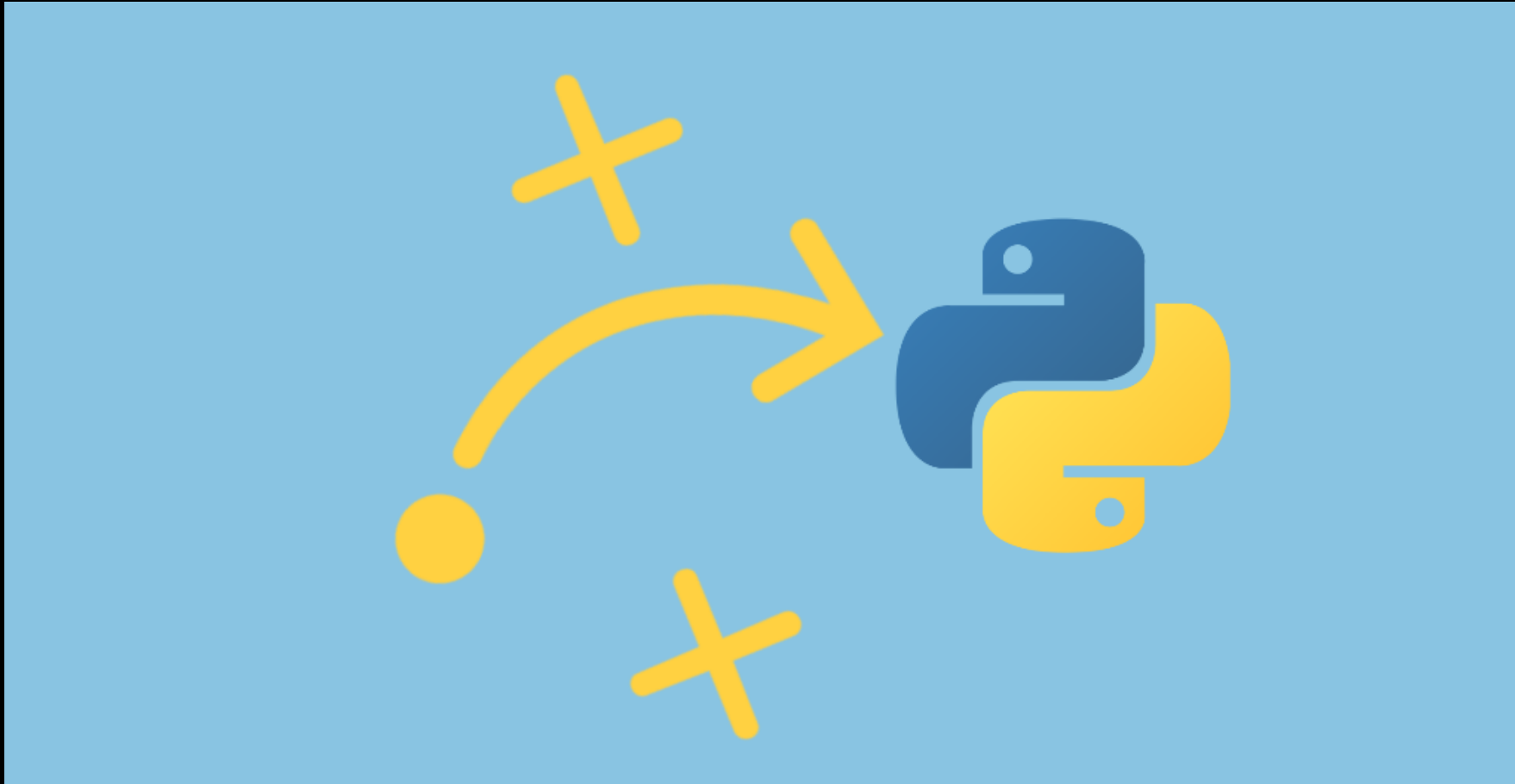
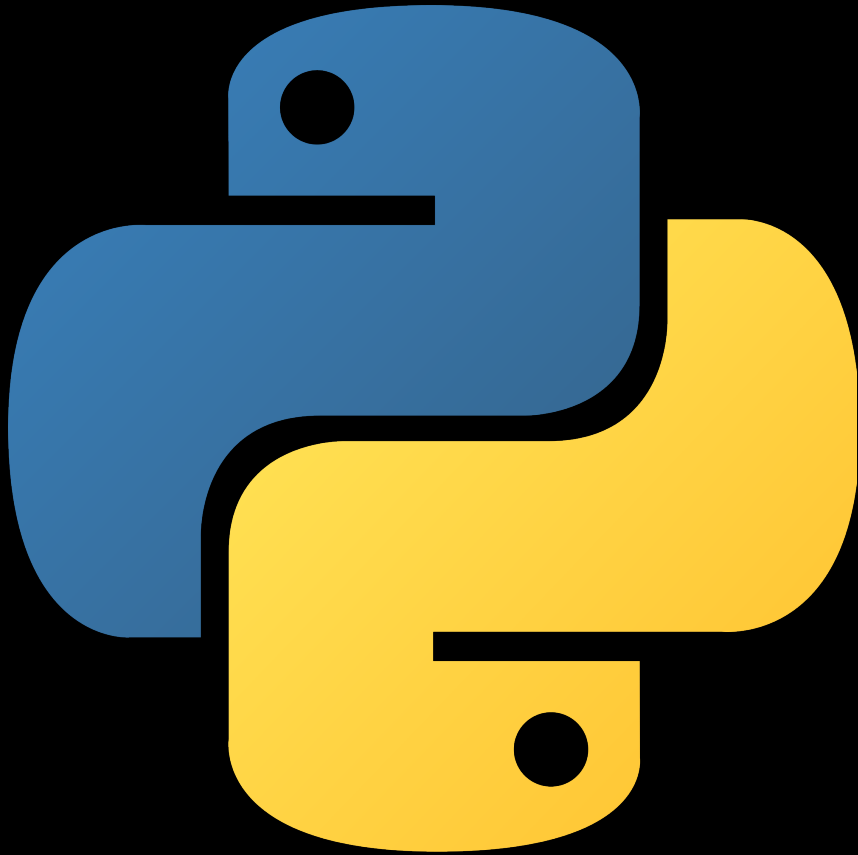


Python vs. ...



Python vs. .NET (or Java)



vs.



Python vs. .NET and C#

	.NET	Python
Open source	Somewhat	Yes
Compiled	Yes (JIT)	Typically no
Owned by a company	Yes	No
Base class library	Yes	Yes
Web app capabilities	Very strong	Very strong
Database capabilities	Very strong	Very strong
Mobile app capabilities	Very strong	Poor
Desktop app capabilities	Very strong	Moderate
Stack overflow Rank	4	1
TIOBE rank	5	3
Price	Free	Free
General purpose language	Yes	Yes
Scientific computing level	Moderate	Very strong

Python vs. MATLAB



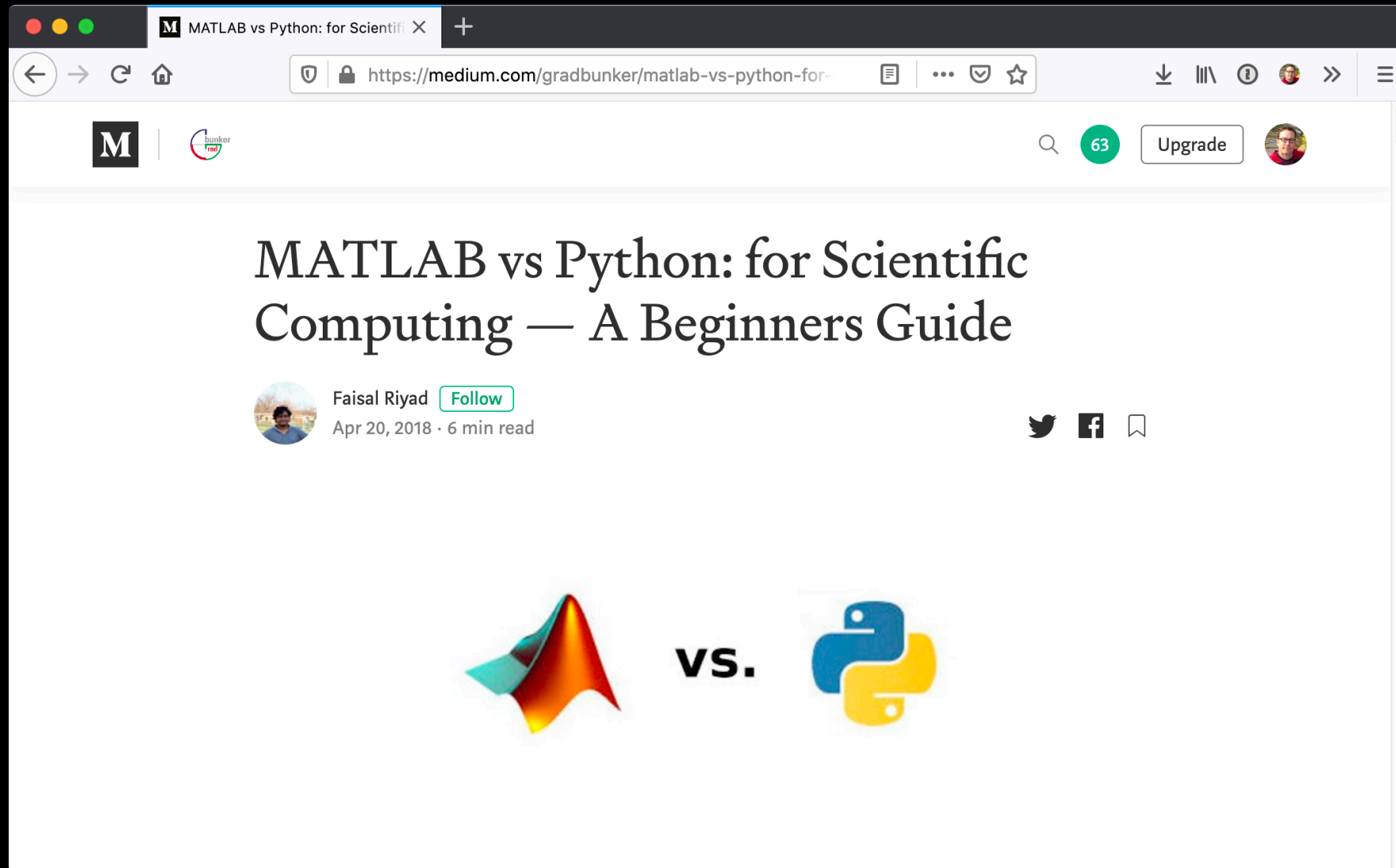
vs.



Python vs. MATLAB

	MATLAB	Python
Open source	No	Yes
Compiled	No	Typically no
Owned by a company	Yes	No
Base class library	Yes	Yes
Web app capabilities	None	Very strong
Database capabilities	None	Very strong
Mobile app capabilities	None	Poor
Desktop app capabilities	None	Moderate
Stack overflow Rank	$-\infty$	1
TIOBE rank	$-\infty$	3
Price	\$860/user/year	Free
General purpose language	No	Yes
Scientific computing level	Strong	Very strong

Python vs. MATLAB



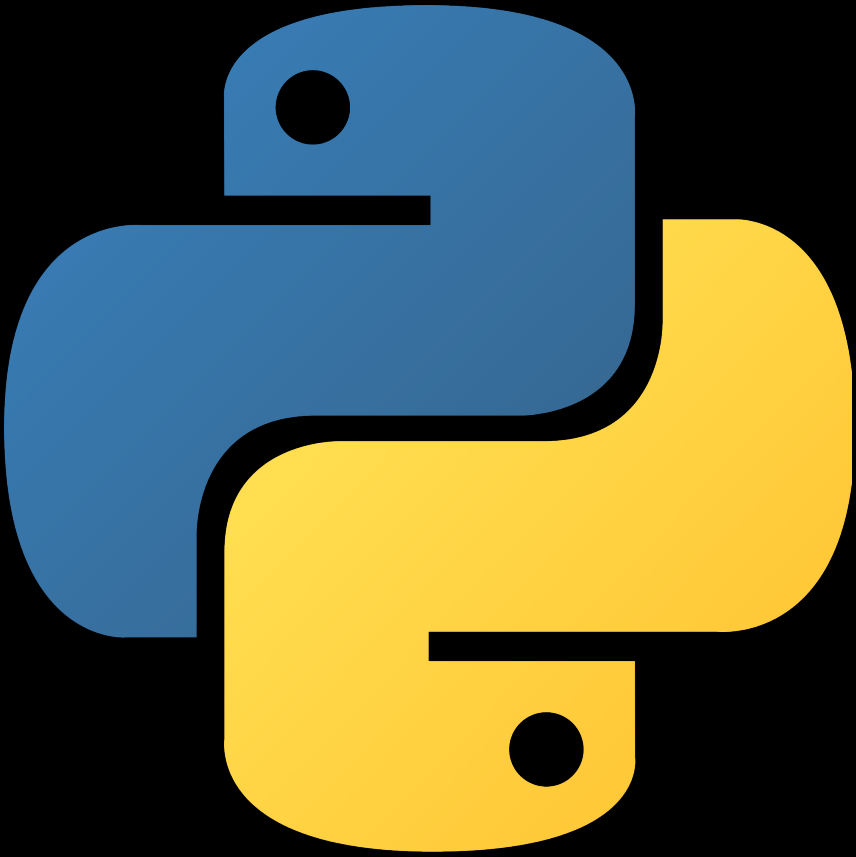
<https://medium.com/gradbunker/matlab-vs-python-for-scientific-computing-a-beginners-guide-a27f4dcbbc81>

Python vs. MATLAB

MATLAB has become a legacy language or tool for the scientific community.

- Python and its companion libraries are getting more and more sophisticated day by day.
- For large scale problems, Python is lot more expressive and readable as compared to MATLAB scripts.
- Python is free and highly adopted
- Has version solid IDEs such as PyCharm
- Python scientific packages have become available with extensive documentation for data visualization, machine learning, natural language processing, complex data analysis and more
- Using Python means you can more easily collaborate with people who don't have access to MATLAB.

Python vs. C/C++



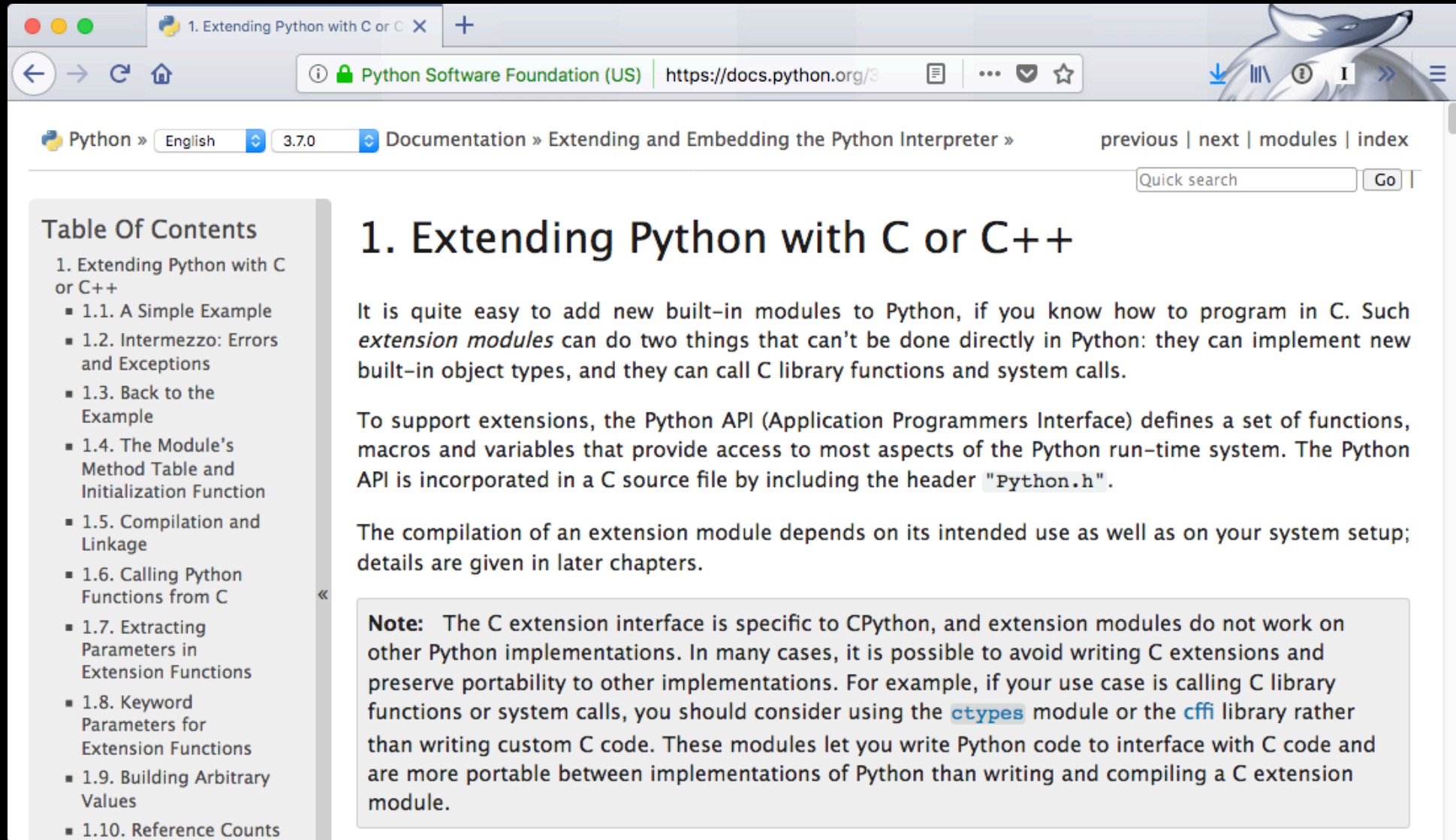
vs.
+



Python vs. C++

	C++	Python
Open source	Yes	Yes
Compiled	Yes	Typically no
Owned by a company	No	No
Base class library	No	Yes
Web app capabilities	None	Very strong
Database capabilities	Moderate	Very strong
Mobile app capabilities	Strong	Poor
Desktop app capabilities	Strong	Moderate
Stack overflow Rank	6	1
TIOBE rank	4	3
Price	Free	Free
General purpose language	Yes	Yes
Scientific computing level	Strong	Very strong

Python has great support for C



The screenshot shows a web browser window displaying the Python Software Foundation documentation. The browser's address bar shows the URL `https://docs.python.org/3`. The page title is "1. Extending Python with C or C++". The page is part of the "3.7.0" documentation, under the "Extending and Embedding the Python Interpreter" section. The page has a search bar and navigation links for "previous", "next", "modules", and "index".

Table Of Contents

- 1. Extending Python with C or C++
 - 1.1. A Simple Example
 - 1.2. Intermezzo: Errors and Exceptions
 - 1.3. Back to the Example
 - 1.4. The Module's Method Table and Initialization Function
 - 1.5. Compilation and Linkage
 - 1.6. Calling Python Functions from C
 - 1.7. Extracting Parameters in Extension Functions
 - 1.8. Keyword Parameters for Extension Functions
 - 1.9. Building Arbitrary Values
 - 1.10. Reference Counts

1. Extending Python with C or C++

It is quite easy to add new built-in modules to Python, if you know how to program in C. Such *extension modules* can do two things that can't be done directly in Python: they can implement new built-in object types, and they can call C library functions and system calls.

To support extensions, the Python API (Application Programmers Interface) defines a set of functions, macros and variables that provide access to most aspects of the Python run-time system. The Python API is incorporated in a C source file by including the header `"Python.h"`.

The compilation of an extension module depends on its intended use as well as on your system setup; details are given in later chapters.

Note: The C extension interface is specific to CPython, and extension modules do not work on other Python implementations. In many cases, it is possible to avoid writing C extensions and preserve portability to other implementations. For example, if your use case is calling C library functions or system calls, you should consider using the `ctypes` module or the `cffi` library rather than writing custom C code. These modules let you write Python code to interface with C code and are more portable between implementations of Python than writing and compiling a C extension module.

What does Cython look like?

```
# Pure Python
import math

def do_math(start: int, num: int):
    dist = 0.0
    pos = start
    k_sq = 1000 * 1000

    while pos < num:
        pos += 1
        dist = math.sqrt((pos - k_sq) * (pos - k_sq))
```

Using C directly from Python

Building a Python C Extension

https://realpython.com/build-python-c-extension-module/

Real Python

Start Here

Learn Python

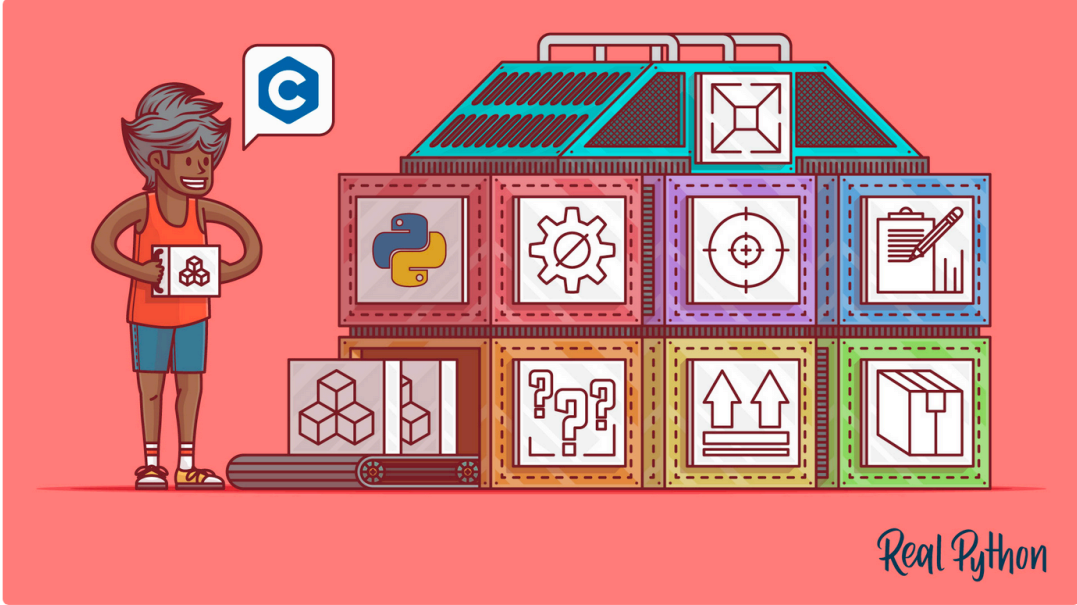
Store

More

Search...

Join

Sign In



Real Python

Building a Python C Extension Module

by Danish Prakash · Oct 07, 2019 · 9 Comments · advanced python

[Tweet](#) [Share](#) [Email](#)

— FREE Email Series —

Python Tricks

```
1 # How to merge two dicts
2 # in Python 3.5+
3
4 >>> x = {'a': 1, 'b': 2}
5 >>> y = {'b': 3, 'c': 4}
6
7 >>> z = {**x, **y}
8
9 >>> z
10 {'c': 4, 'a': 1, 'b': 3}
```

Email...

Get Python Tricks »

No spam. Unsubscribe any time.

Improve Your Python

<https://realpython.com/build-python-c-extension-module/>