



Indian Pythonista

# Brief history of python

- Created by Guido Van Rossum



- Idea generated in late 1980s
- Implementation began in December 1989
- First released in 1991
- Python 2: 2000
- Python 3: 2008

# **Features of python**

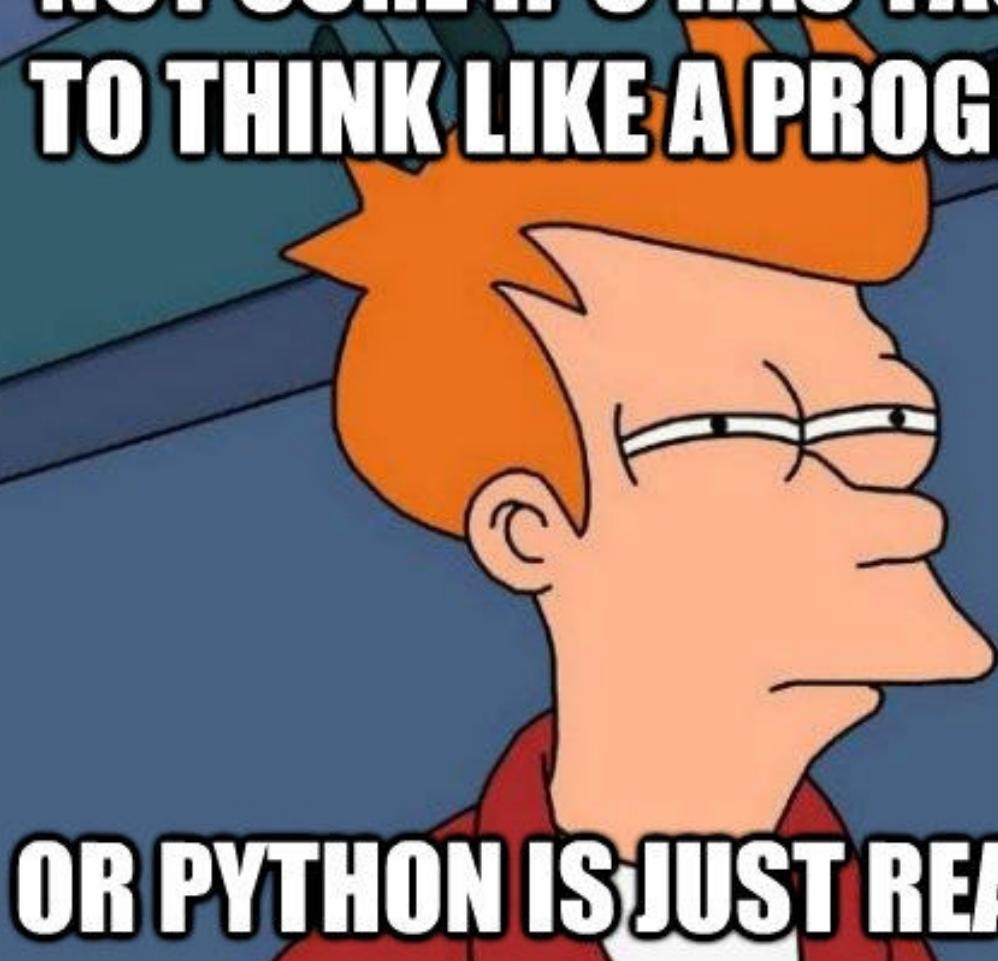
## **1. Simple**

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

## **2. Easy to Learn**

As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

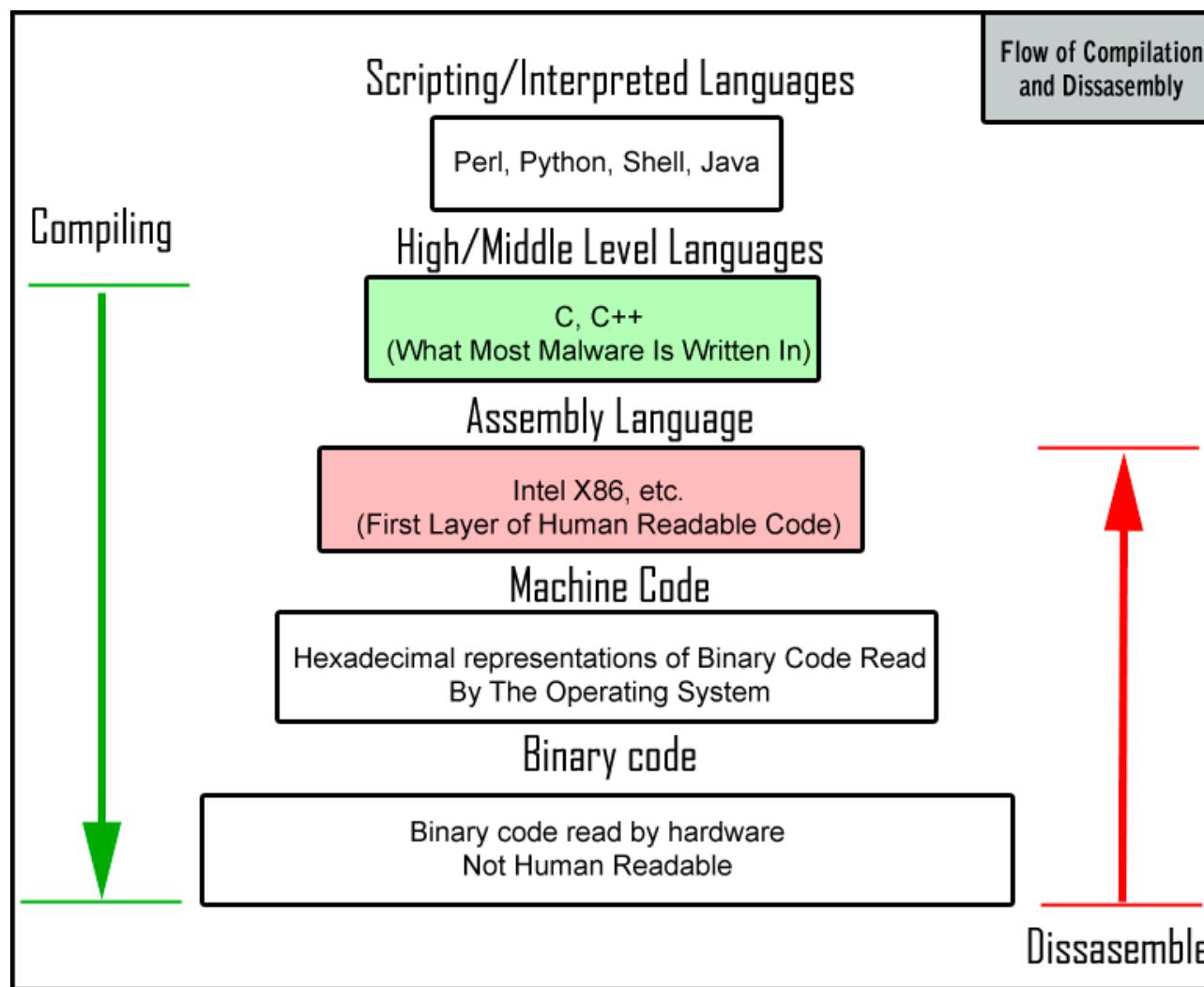
**NOT SURE IF C HAS TAUGHT ME  
TO THINK LIKE A PROGRAMMER**



**OR PYTHON IS JUST REALLY EASY**

# 3. High level

When you write programs in Python, you never need to bother about the low-level details such as managing the memory used by your program, etc.



# 4. Interpreted

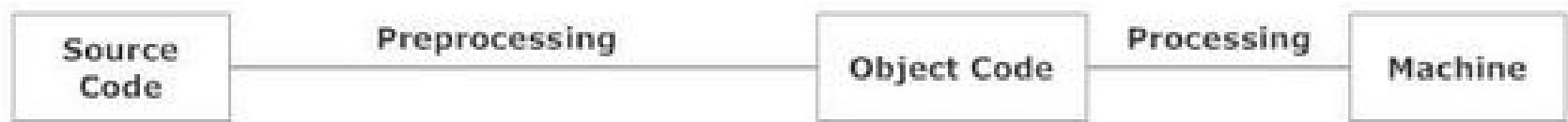


Figure: Compiler



Figure: Interpreter

## Interpreter

Translates program one statement at a time.

It takes less amount of time to analyze the source code but the overall execution time is slower.

No intermediate object code is generated, hence are memory efficient.

Continues translating the program until the first error is met, in which case it stops. Hence debugging is easy.

Programming language like Python, Ruby use interpreters.

## Compiler

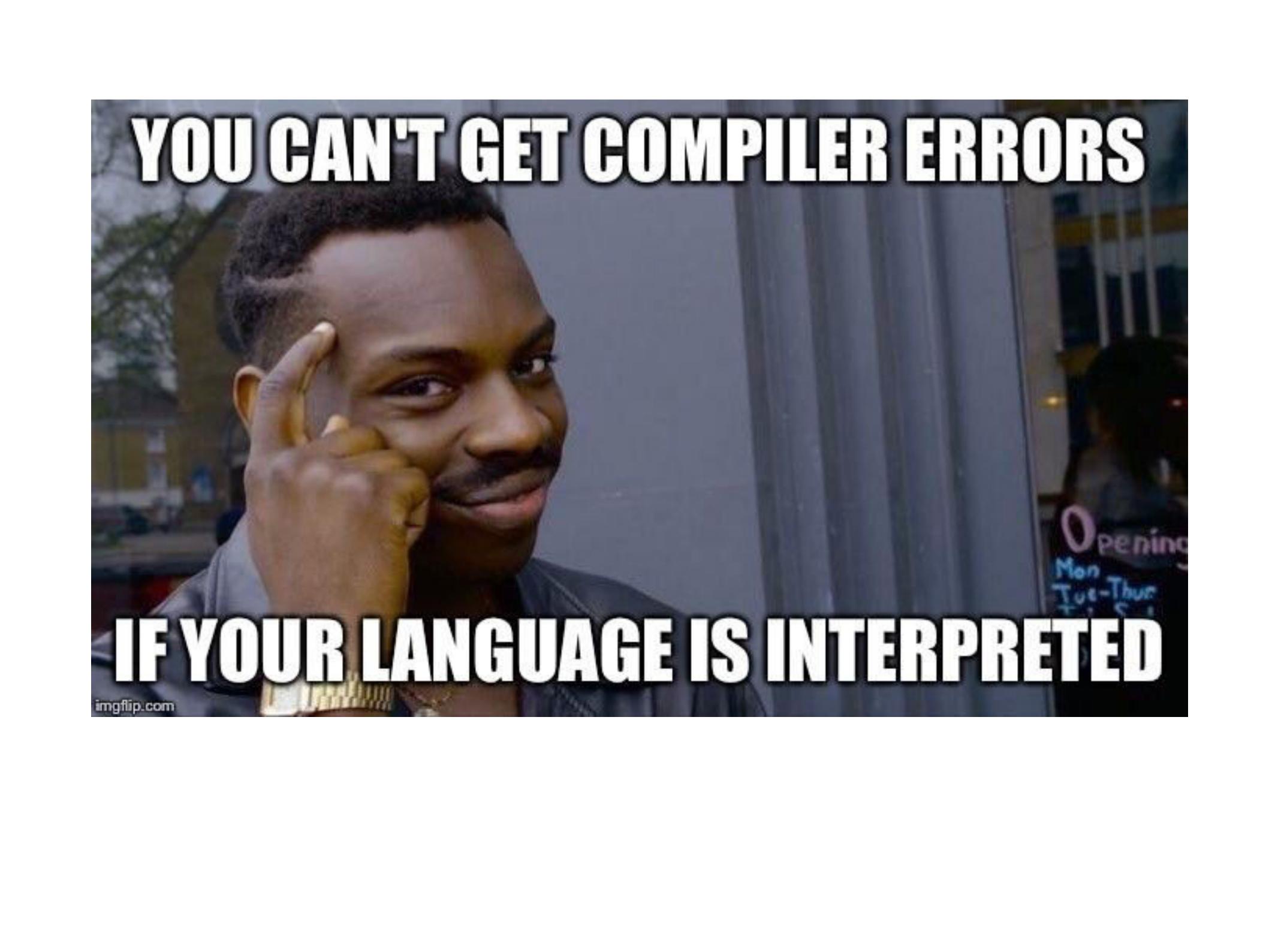
Scans the entire program and translates it as a whole into machine code.

It takes large amount of time to analyze the source code but the overall execution time is comparatively faster.

Generates intermediate object code which further requires linking, hence requires more memory.

It generates the error message only after scanning the whole program. Hence debugging is comparatively hard.

Programming language like C, C++ use compilers.

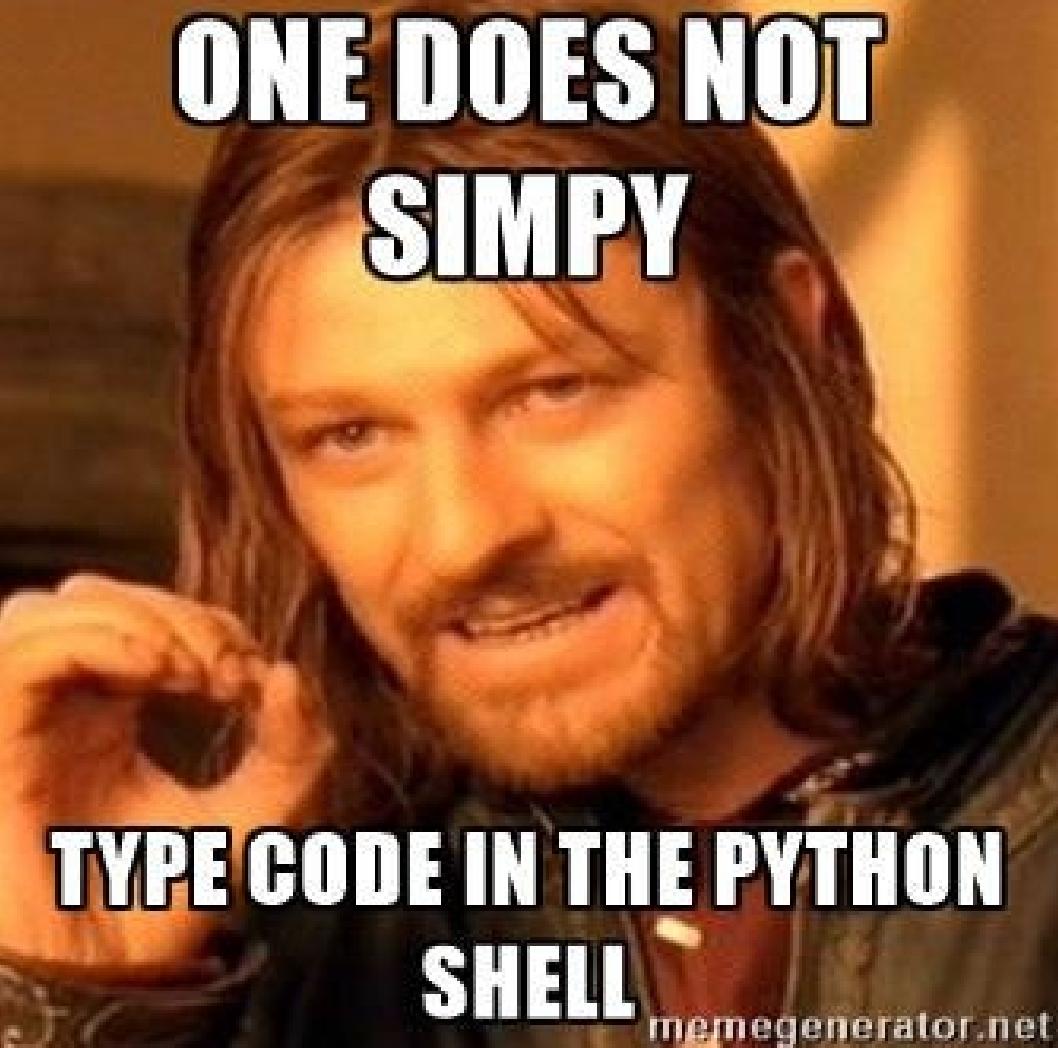


**YOU CAN'T GET COMPILER ERRORS**

**IF YOUR LANGUAGE IS INTERPRETED**

# 5. Interactive

```
x - Terminal File Edit View Search Terminal Help
nikhil@nikhil:~$ python3
Python 3.5.2+ (default, Sep 22 2016, 12:18:14)
[GCC 6.2.0 20160927] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 1 + 1
2
>>> print("OMG!!!")
OMG!!!
>>> █
```

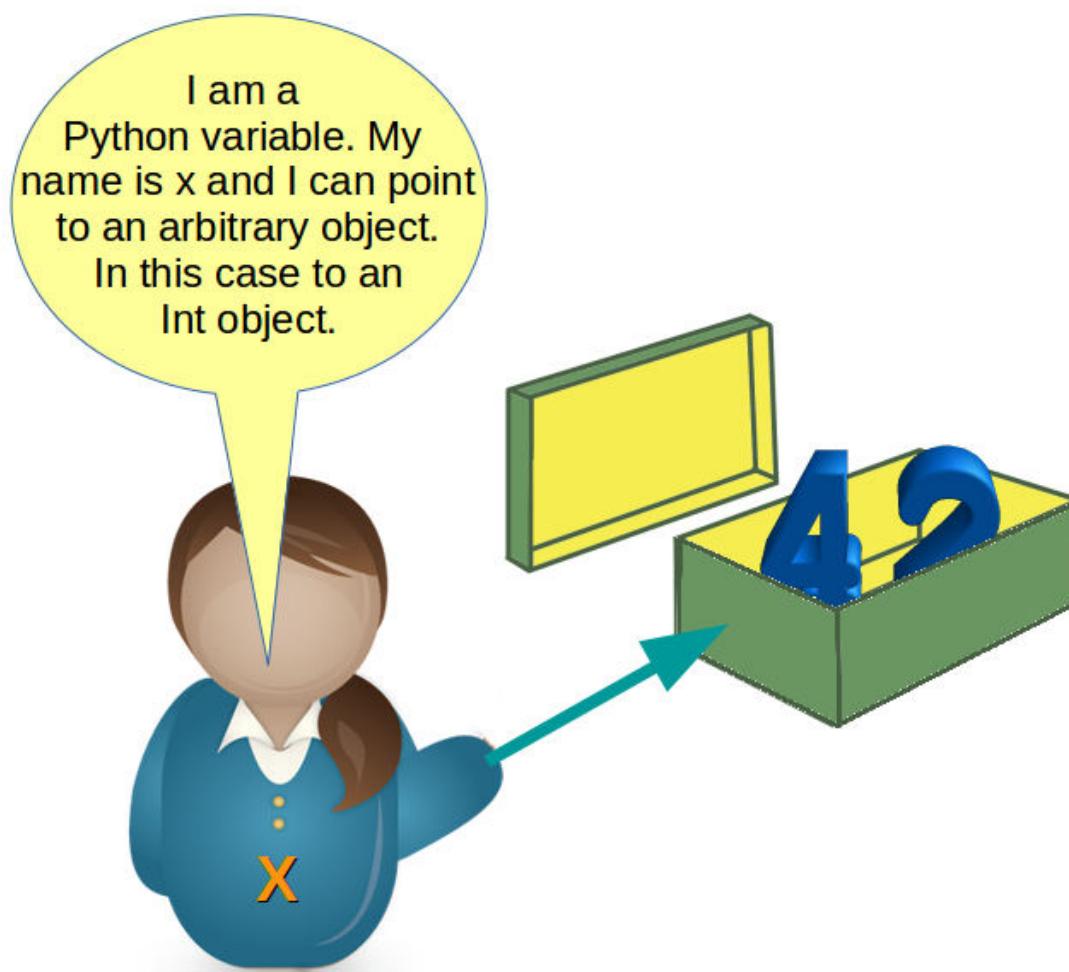


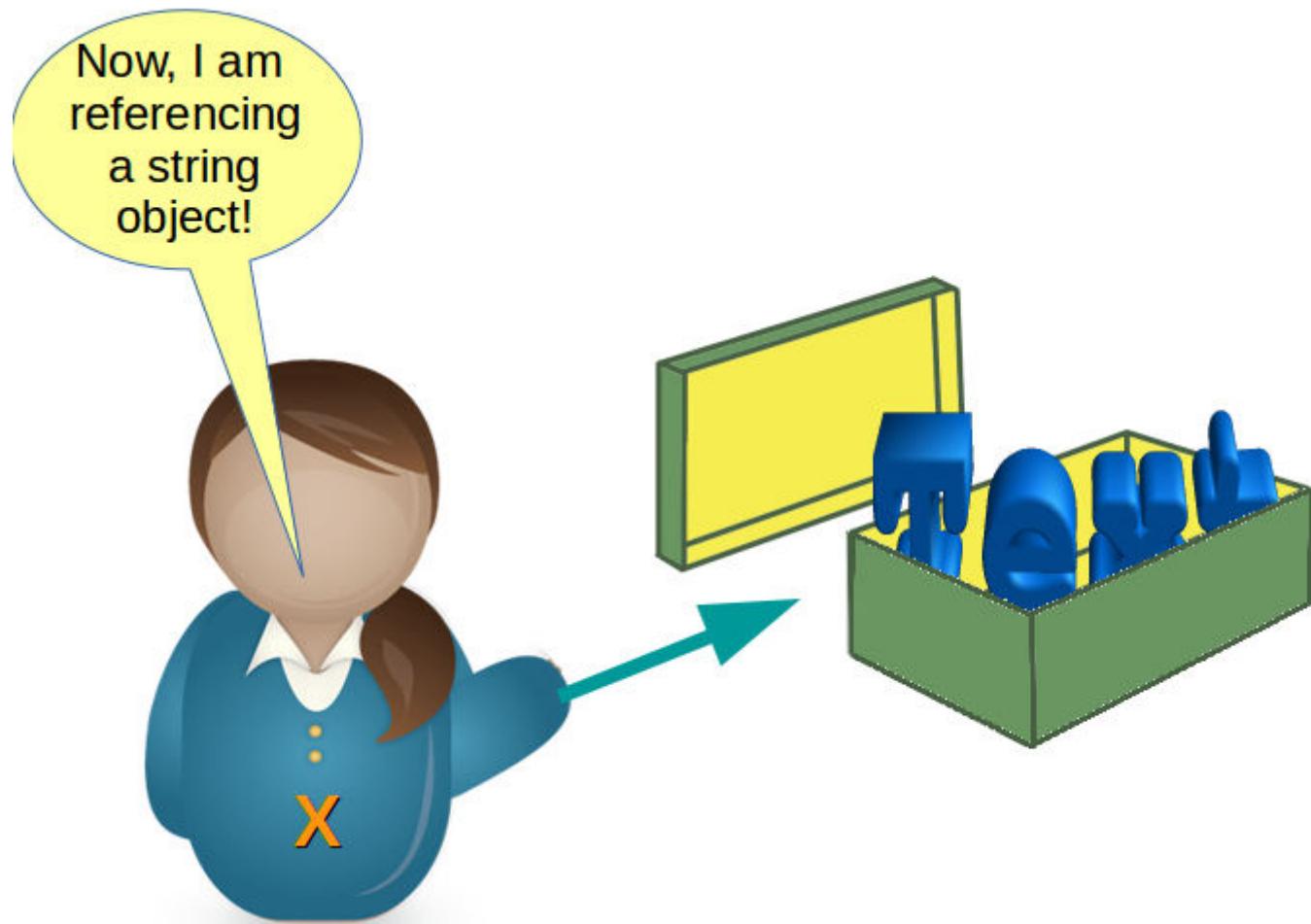
**ONE DOES NOT  
SIMPLY**

**TYPE CODE IN THE PYTHON  
SHELL**

[memegenerator.net](http://memegenerator.net)

# 6. Dynamically typed



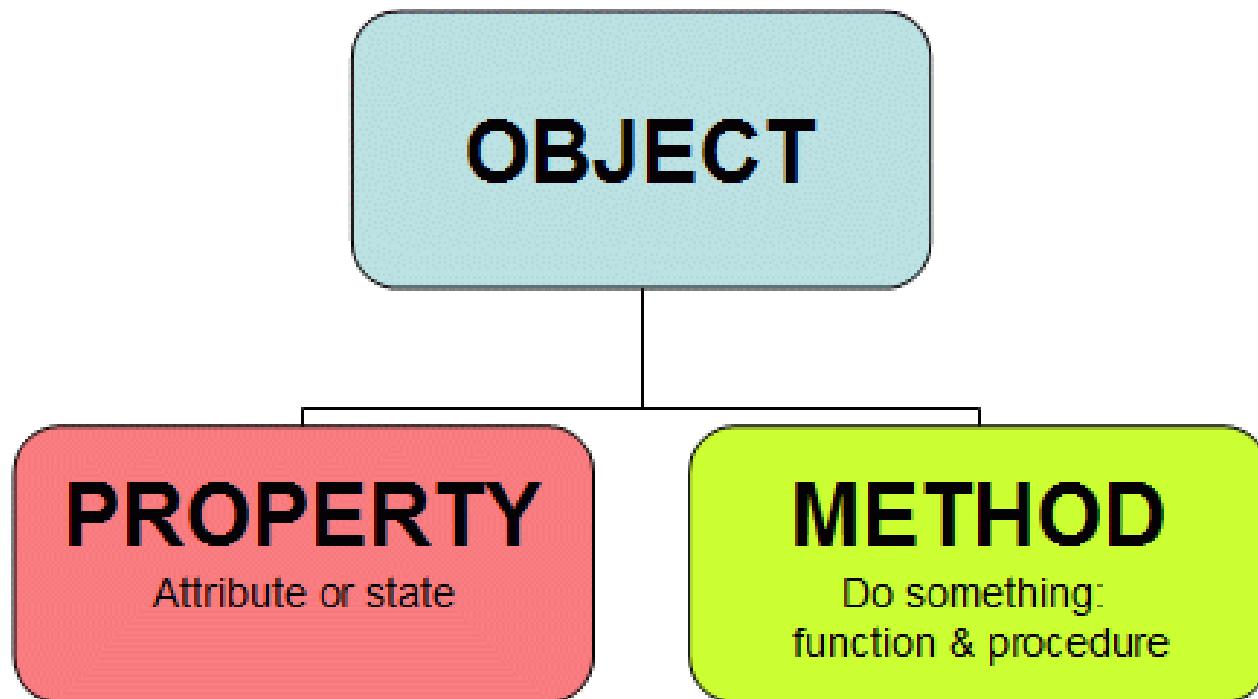


**YOU DON'T HAVE TO FIX TYPE  
ERRORS**

**IF YOU USE A DYNAMIC  
LANGUAGE**

memegenerator.net

# 7. Object oriented

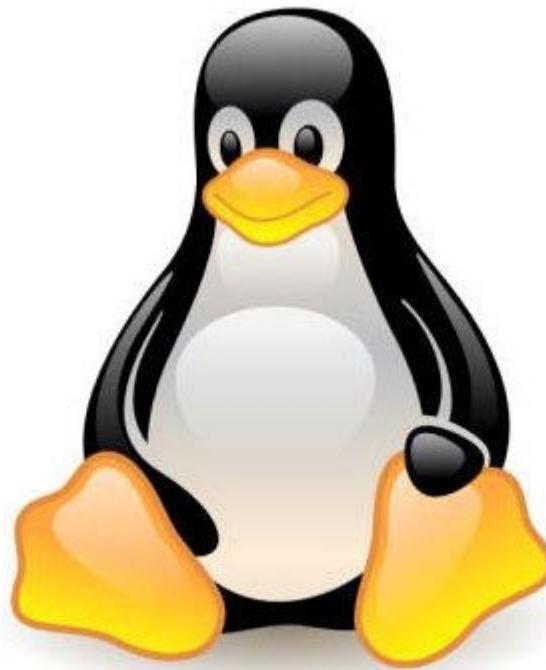
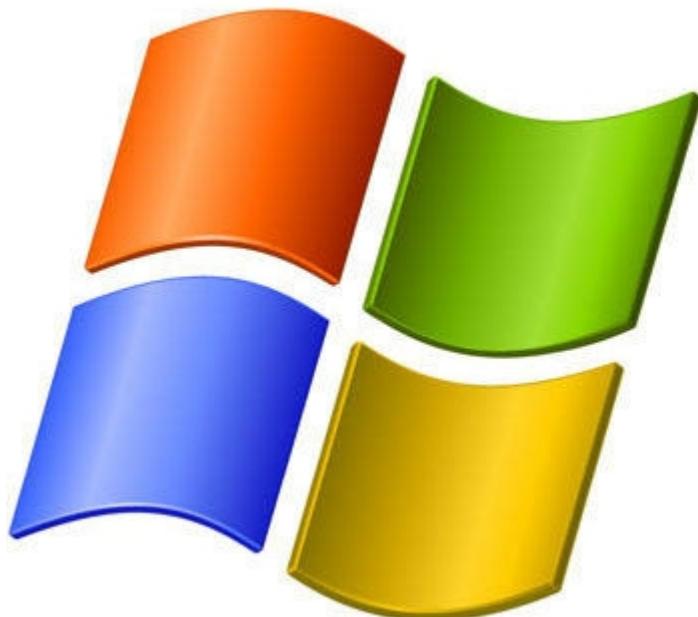


I DON'T ALWAYS WRITE  
CODE



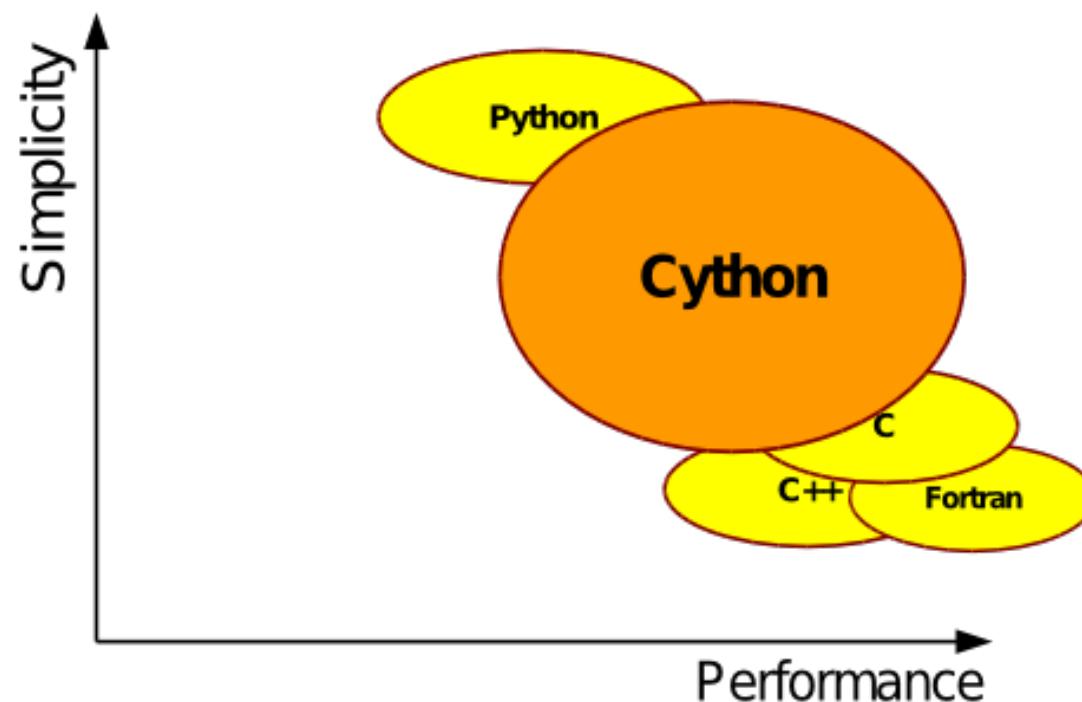
BUT WHEN I DO, I PREFER  
OBJECT-ORIENTED  
PROGRAMMING

# 8. Portable



# 9. Extensible

If you need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, you can code that part of your program in C or C++ and then use them from your Python program.



# Different flavors....



## Python Flavors

### Python Implementations

*"Why uses a alternative Python implementation? To be able to uses all the stuff regarding this technology stack, for example using Jython it is possible use all the Java stuff and Libraries."*

Implementation	Virtual Machine	Language
CPython	CPythonVM	C
Jython	JVM	Java
IronPython	CLR	C#
Brython	Javascript Interpreter(V8)	Javascript
RubyPython	RubyVM	Ruby



**WE GOT ALL KIND OF  
PYTHON**

**CPYTHON, JYTHON, IRONPYTHON, PYPY, PYSTON, MYPY,  
CYTHON**

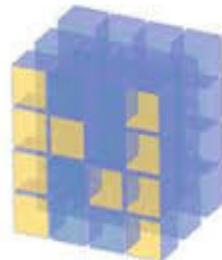
[memegenerator.net](http://memegenerator.net)

# 10. Extensive libraries

There are many python libraries and you can make and publish one yourself!



Requests



NumPy



# PIP INSTALL



## ALL THE THINGS!

memegenerator.net

# Applications of python

- Web development (frameworks like django and flask)
- Data analysis (scientific, numeric computing and data visualization)
- Desktop GUIs
- Automating tasks

**WRITES 10 LINE  
PYTHON SCRIPT**



**ELIMINATES THE  
NEED FOR 5 JOBS**

# BIG QUESTION

PYTHON 2 OR PYTHON 3?

WTF!

# Python 2 vs Python 3

According to PSF,

> Python 2.x is legacy, Python 3.x is the present and future of the language.

Python 3 was designed to rectify certain fundamental design flaws in the language.

But it was realised that it was not possible to implement all changes while retaining full backwards compatibility.

> The guiding principle of Python 3 was: "reduce feature duplication by removing old ways of doing things"

Python 2 is gonna retire in 2020

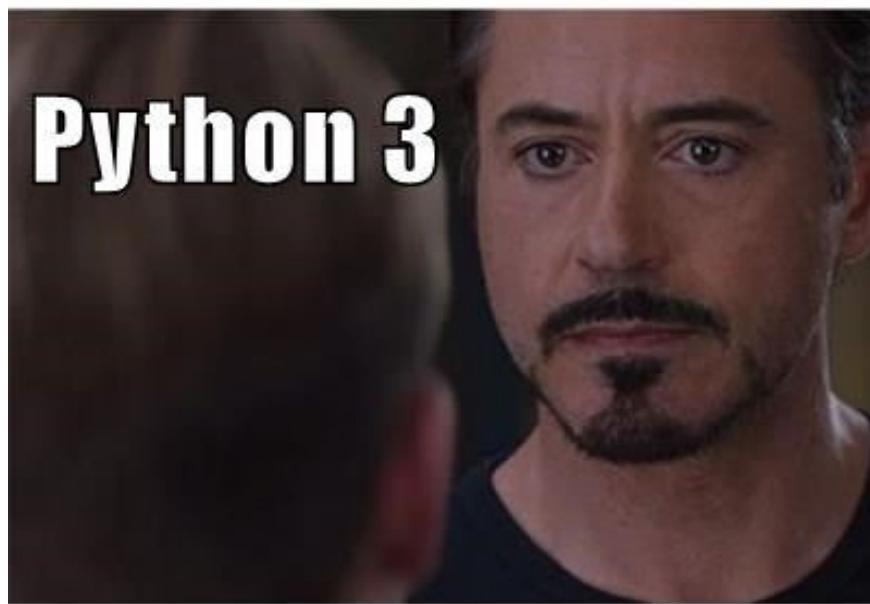
<https://pythonclock.org>



**Python 2**

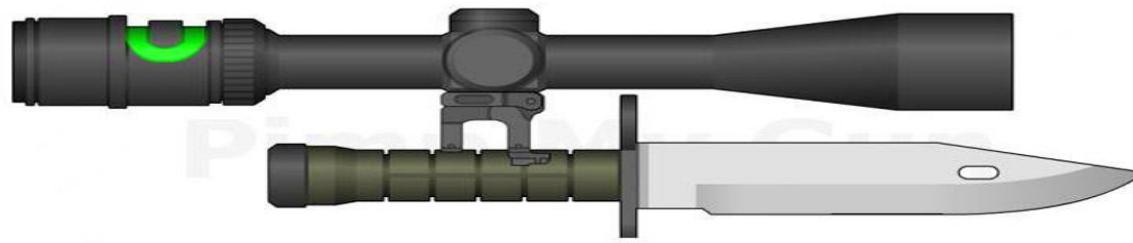


**Python 3**



MARVEL  
**CIVIL WAR**  
CAPTAIN AMERICA  
MAY 6, 2016

# Assembler



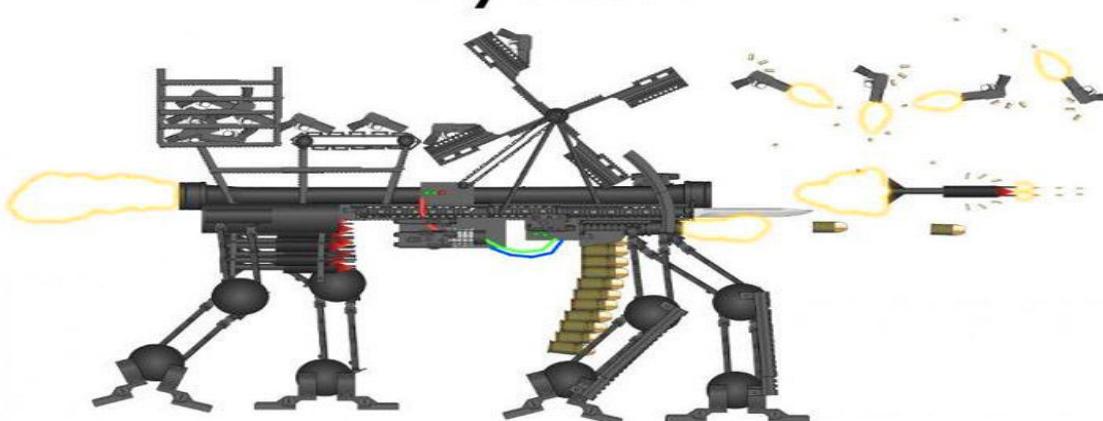
C



C++



Python



**ARE YOU**

**READY?**

**ARE YOU SURE**

**YOU'RE READY?**



# Only if you use windows...

