Why Python?

- Easy to read, learn and code
- Dynamic Typing
- Free, Open Source
- Portable
- Extensive Third-Party Libraries
- Interpreted Language
- Functional, Object-Oriented, and Procedural
- Improved Productivity

Disadvantages of Python

- Slow at runtime
- High memory consumption
- Runtime Errors
- Not used in the Enterprise Development
- Simplicity

helloworld

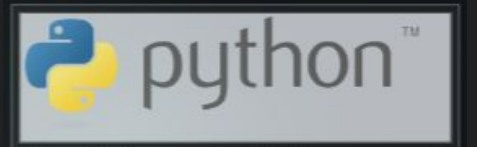
Input: QwQ

lang	code	time
c	1.c	1.0ms
c	1.c	1.3ms
c	1.c	1.4ms
python	1.py	12ms
python	1.py	14ms
python	1.py	28ms

History

- Version 1 (1994)
- Version 2 (2000)
- Version 3 (2008)

Old Python logo, 1990s–2006



New Python logo, 2006–present


Guido van Rossum in 2014

The Zen of Python

```
1 import this
```

https://cs50.harvard.edu/python/2022/
https://cs50.edx.org/python

CS50 Certificate

CS50 congratulates

Saman Khodarahmi

on completion of CS50's Introduction to Programming with Python, including nine problem sets and one final project.

Awarded from Cambridge, Massachusetts, in the year two thousand twenty-four.

David J. Malan
Gordon McKay Professor of the Practice of Computer Science
Harvard University

https://cs50.harvard.edu/certificates/abba9d16-a511-44c2-9ea5-239e907b3fb0

Problem Set 3

What to Do

1. Log into cs50.dev, which is a cloud-based v
course pre-installed. No need to download
2. Execute `update50` in your codespace's term
3. Submit all of the problems below:
   - Fuel Gauge
   - Felipe's Taqueria
   - Grocery List
   - Outdated

Contact me

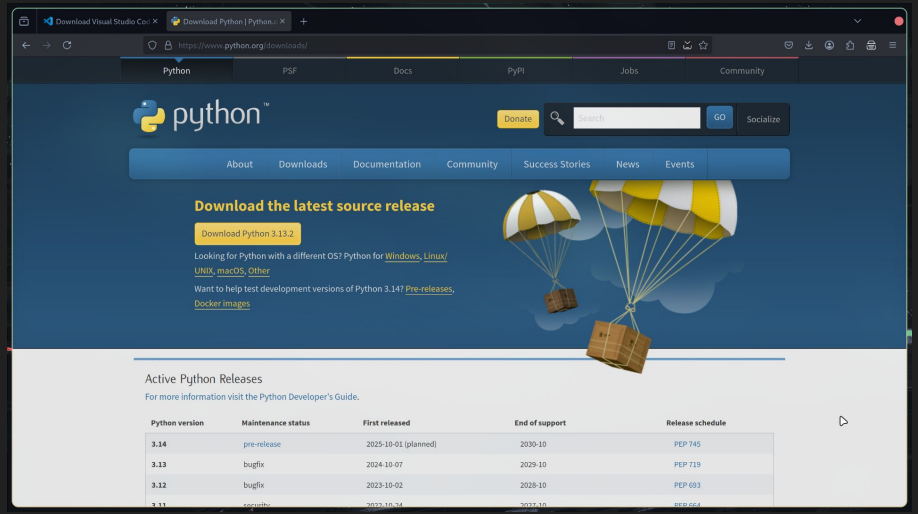
Telegram: @Neutrallin

Email: neutralobs@gmail.com

https://github.com/Neutrallin4/PythonClass

https://code.visualstudio.com/download

https://www.python.org/downloads/



Clean Code in Python

- Easy to understand
- More efficient
- Easier to maintain, scale, debug, and refactor

https://testdriven.io/blog/clean-code-python/

https://peps.python.org/pep-0008/ PEP 8 (Python Enhancement Proposal)

Documentation

https://docs.python.org/3/library/

```
print(*objects, sep=' ', end='\n', file=None, flush=False)
```

Print *objects* to the text stream *file*, separated by *sep* and followed by *end*. *sep*, *end*, *file*, and *flush*, if present, must be given as keyword arguments.

All non-keyword arguments are converted to strings like `str()` does and written to the stream, separated by *sep* and followed by *end*. Both *sep* and *end* must be strings; they can also be `None`, which means to use the default values. If no *objects* are given, `print()` will just write *end*.

The *file* argument must be an object with a `write(string)` method; if it is not present or `None`, `sys.stdout` will be used. Since printed arguments are converted to text strings, `print()` cannot be used with binary mode file objects. For these, use `file.write(...)` instead.

Output buffering is usually determined by *file*. However, if *flush* is true, the stream is forcibly flushed.

> *Changed in version 3.3:* Added the *flush* keyword argument.