

Module-03

Comprehensive Guide to Python

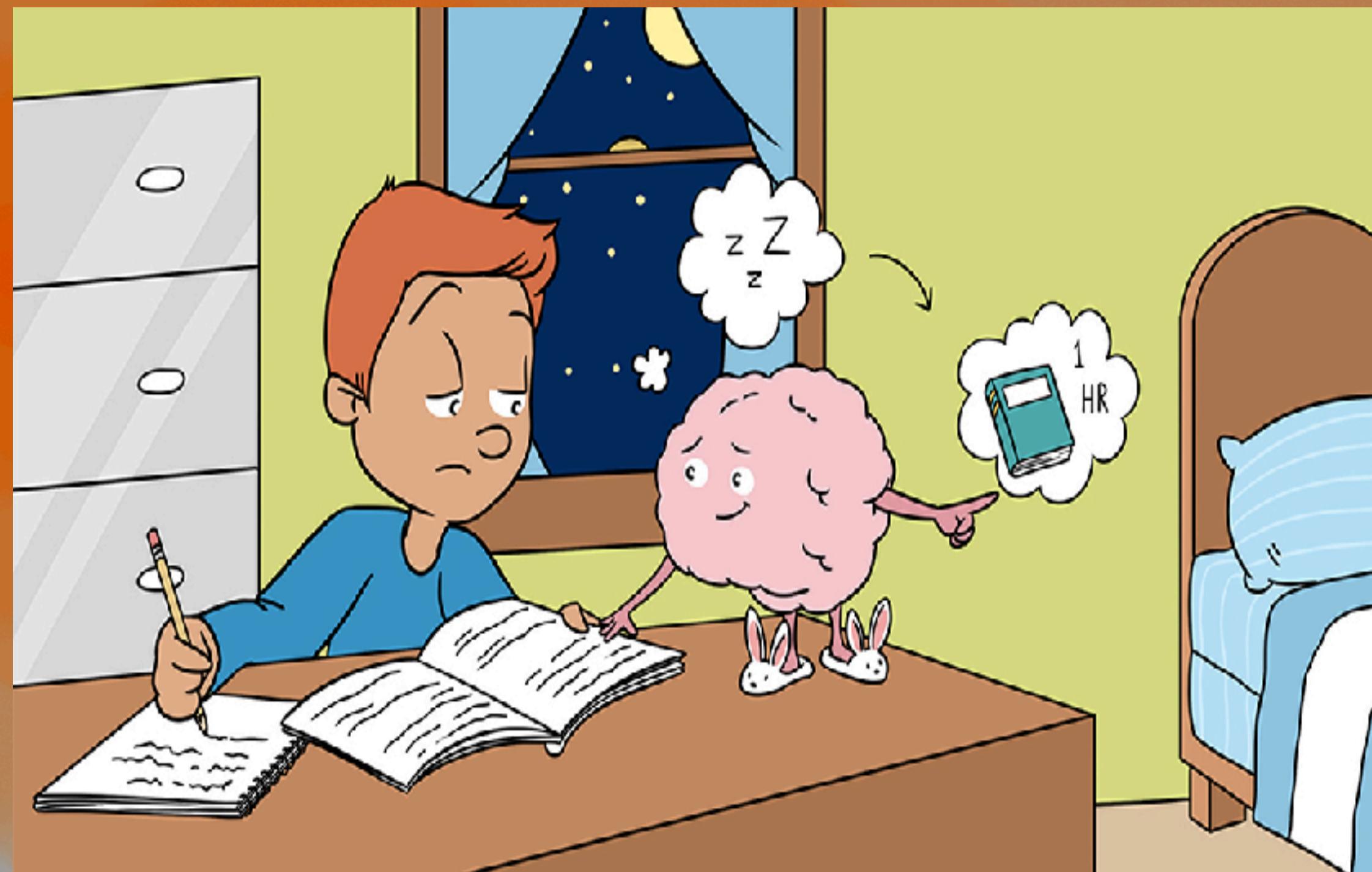


Contents

- ▶ **Studying Effectively.**
- ▶ **What is Python.**
- ▶ **Why Python.**
- ▶ **Interpreted VS Compiled Language.**
- ▶ **History of Python.**
- ▶ **Setting Up Programming Environment.**
- ▶ **How Python Code is Being Executed?**

Studying Effectively

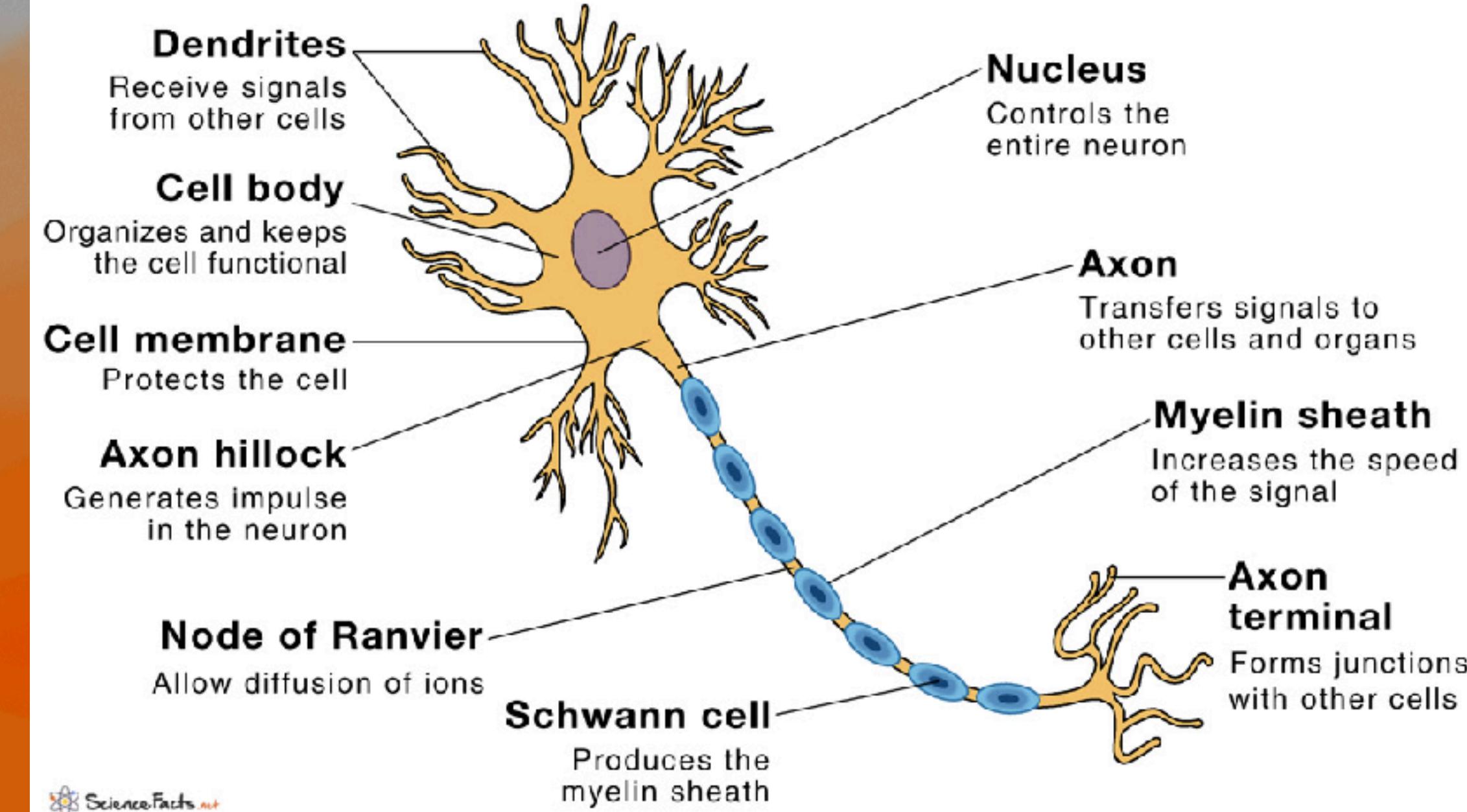
- ▶ Don't study for too long if you are not enjoying it. Take short breaks after 20-30 minutes.
- ▶ Don't study in the living room, bedroom, etc. Study in a separate place dedicated to studying.
- ▶ More active you are more effective you study. Don't try to memorize without understanding. Try to understand the concept first.
- ▶ Don't always study alone. Studying in groups helps a lot.
- ▶ Don't highlight text blindly. Highlighting doesn't help that much. It only indicates Recognition not Recollection of the topic.



Studying Effectively...

- ▶ Always take notes. Reviewing the notes after a short time helps a lot.
- ▶ Always try to teach others what you have learned. Teaching is the best way of learning.
- ▶ Sleep is so much important for pushing something into your long-term memory. Get at least 7-8 hours of sound sleep.
- ▶ If studying from book then use the SQ3R(survey, question, recite, read, review) method while studying.
- ▶ Use Mnemonics. It's the best way to memorize facts.

Parts of a Neuron with Functions



What Is Python?

- ▶ Python is a high-level, computer programming language.
- ▶ A programming language is a formal language that is used to write instructions or code that can be executed by a computer.
- ▶ The primary purpose of a programming language is to allow programmers to communicate with computers and give them specific instructions or commands to perform various tasks.



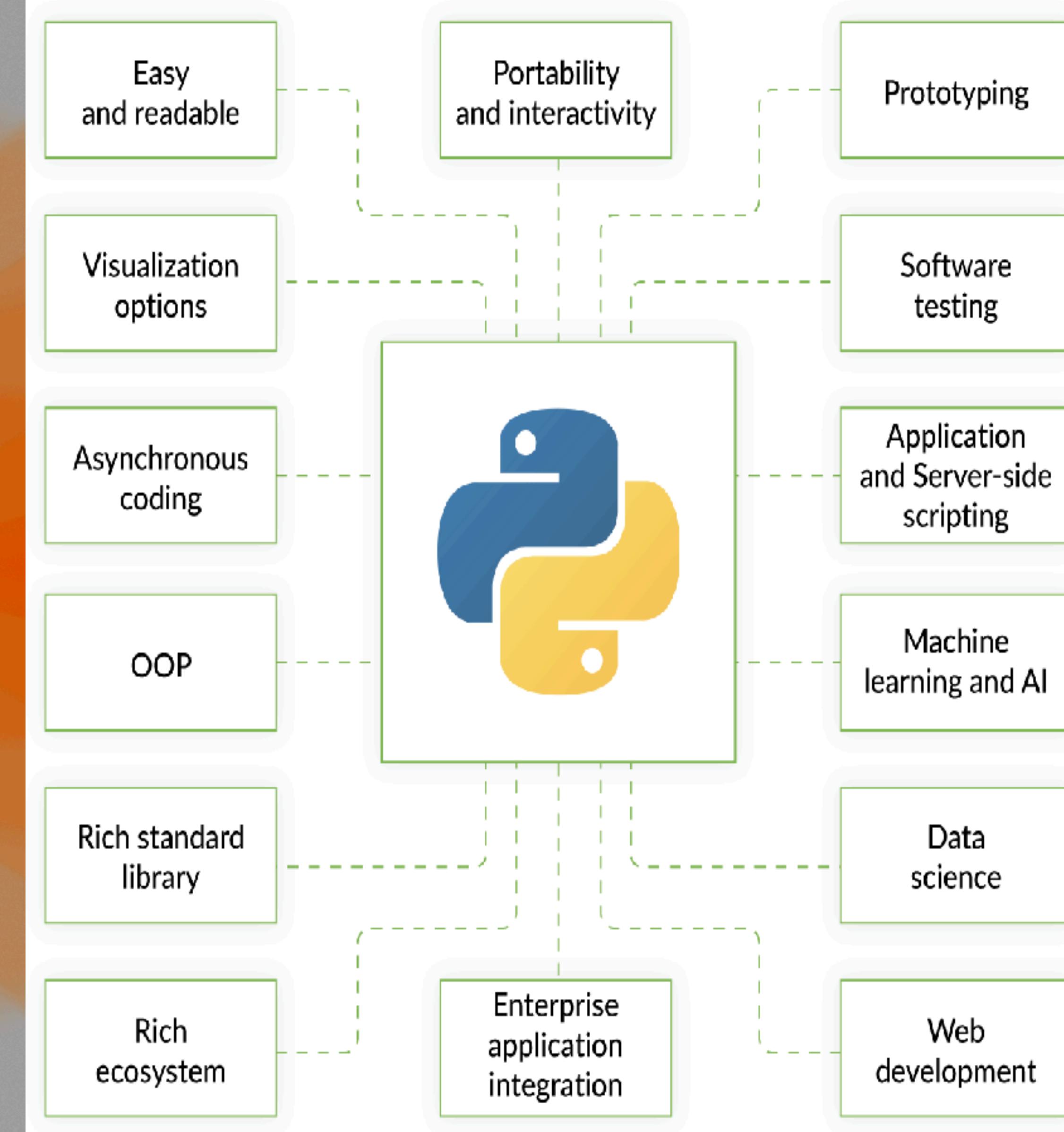
Why Python?

- ▶ **Versatility:** The versatility of Python makes it a popular choice among developers, scientists, and researchers who need a programming language that can be used for a wide range of tasks and applications, such as:
- ▶ **Web development:** Python has many frameworks and libraries that make it an excellent choice for building web applications.
- ▶ **Scientific computing:** Python has many libraries, such as NumPy, SciPy, and Pandas, that make it an excellent choice for scientific computing and data analysis.



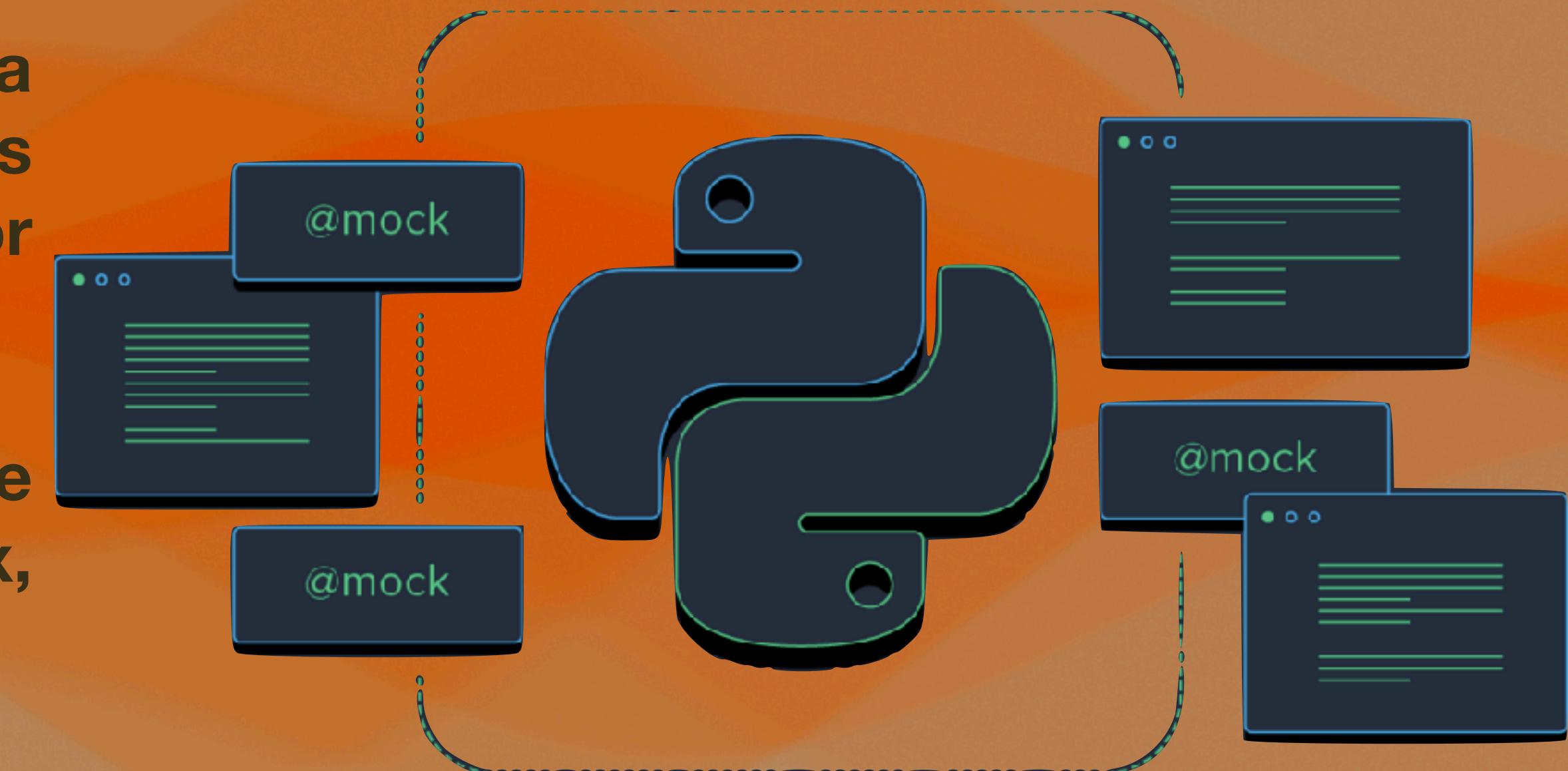
Why Python?...

- ▶ **Scripting:** Python is a great choice for scripting tasks, such as automating routine tasks and batch processing.
- ▶ **Game development:** Python has many libraries, such as Pygame and PyOpenGL, that make it an excellent choice for game development.
- ▶ **Desktop application development:** Python has many libraries, such as Tkinter and PyQt, that make it an excellent choice for building desktop applications.
- ▶ **Machine learning:** Python has many libraries, such as TensorFlow, PyTorch, and Scikit-learn, that make it an excellent choice for machine learning and artificial intelligence applications.



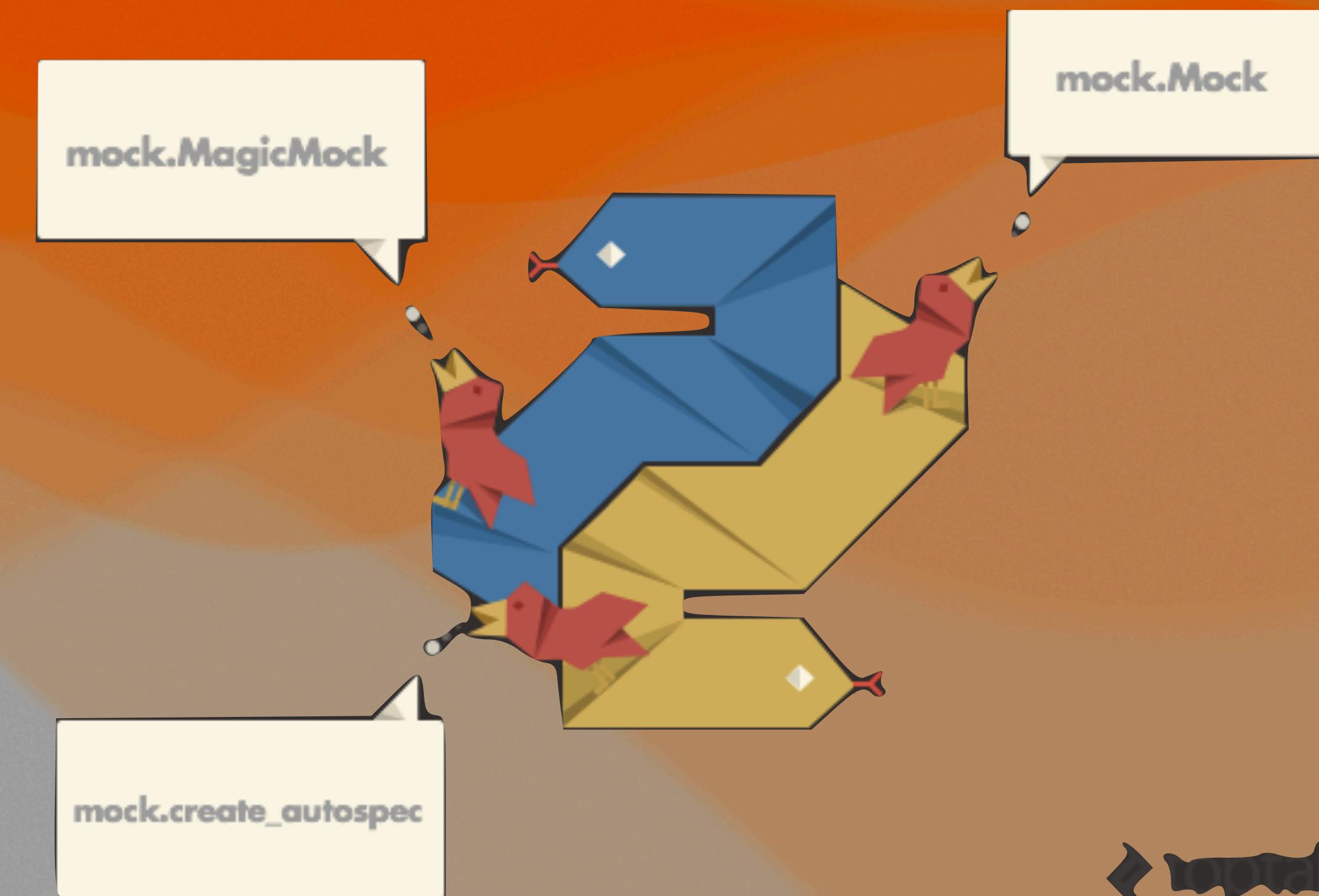
Why Python?...

- ▶ **Readability:** Python has a simple and easy-to-read syntax, which makes it easy for developers to understand and write code.
- ▶ **Large standard library:** Python comes with a large standard library, which provides developers with many useful modules and functions for tasks like file I/O, networking, and more.
- ▶ **Cross-platform:** Python can run on multiple platforms, including Windows, macOS, and Linux, making it very versatile.
- ▶ **Open-source:** Python is free and open-source, which means anyone can use and modify it.



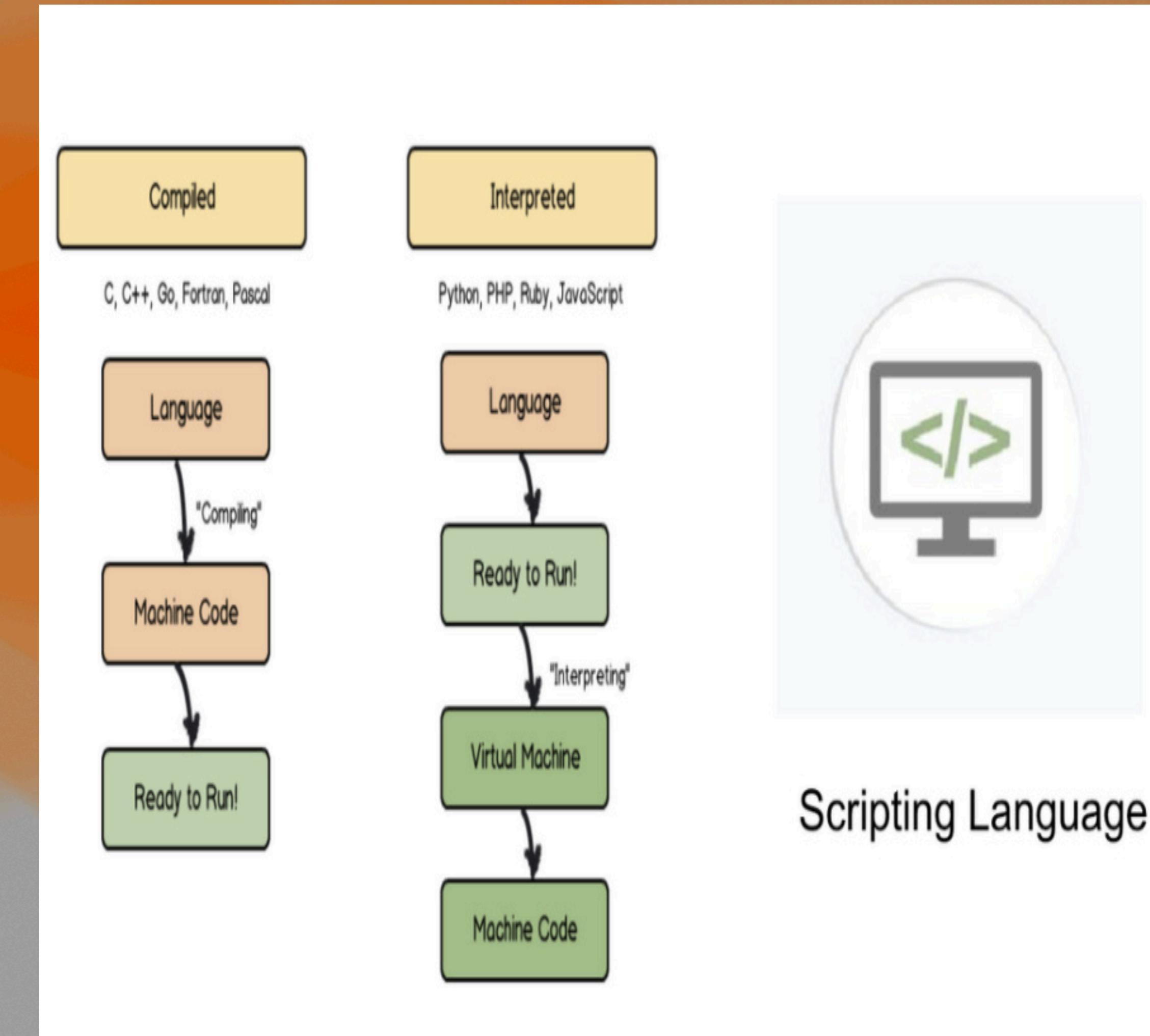
Why Python?...

- ▶ **Interpreted:** Python is an interpreted language, meaning that code can be executed directly without compiling, which makes it faster to write and test code.
- ▶ **Object-oriented:** Python is an object-oriented language, which means that it supports object-oriented programming (OOP) principles such as inheritance, encapsulation, and polymorphism.
- ▶ **Large and active community:** Python has a large and active community, which means there is a lot of support available for developers, including forums, online tutorials, and open-source projects.
- ▶ **Third-party libraries:** Python has a vast array of third-party libraries and frameworks, which makes it very flexible and useful for a wide range of tasks, including web development, scientific computing, data analysis, and machine learning.



Interpreted VS Compiled Language

- ▶ A compiled language is a programming language that is converted into machine code so that the processor can execute it. The compiled languages are usually compiled, not interpreted. For better understanding you can go through the types of compiled language – CLEO, COBOL, C, C++, C#, etc.
- ▶ An interpreted language is also a programming language that is commonly interpreted. In this, the implementations perform instructions directly and easily, without compiling a program into machine-language instructions. For better understanding, you can go through the types of the interpreted languages: Python, BASIC, JavaScript, Perl, etc.



Interpreted VS Compiled Language...

S.NO	Compiled Language	Interpreted Language
1	Compiled language follows at least two levels to get from source code to execution.	Interpreted language follows one step to get from source code to execution.
2	A compiled language is converted into machine code so that the processor can execute it.	An interpreted language is a language in which the implementations execute instructions directly without earlier compiling a program into machine language.
4	The compiled programs run faster than interpreted programs.	The interpreted programs run slower than the compiled program.
5	In a compiled language, the code can be executed by the CPU.	In Interpreted languages, the program cannot be compiled, it is interpreted.
6	This language delivers better performance.	This language delivers slower performance.

History of Python

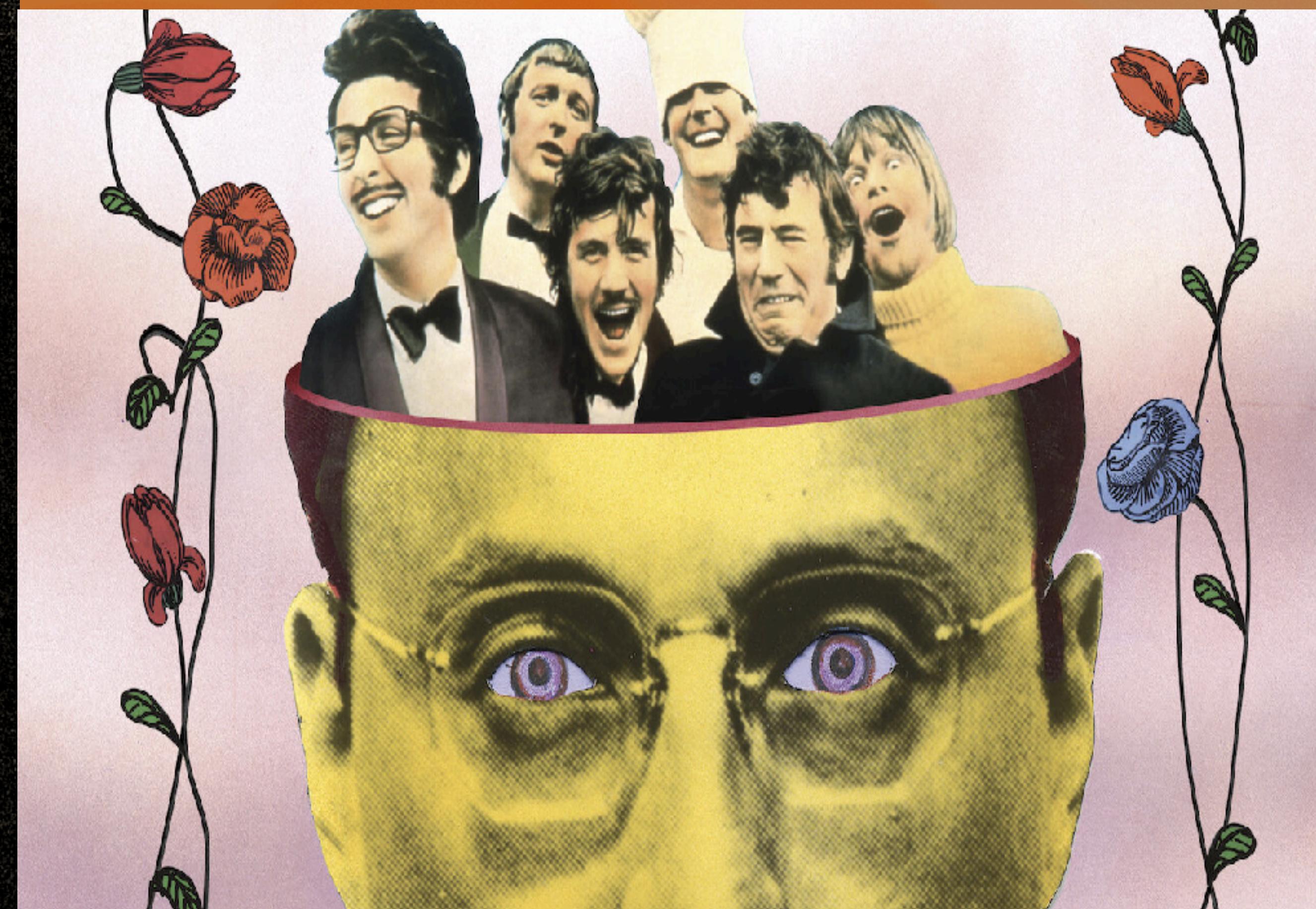
- ▶ Python was first created by Guido van Rossum in the late 1980s, while he was working at the National Research Institute for Mathematics and Computer Science in the Netherlands. Guido wanted to create a language that was easy to read and write, and that would appeal to a broad range of programmers.
- ▶ The first version of Python, version 0.9.0, was released in February 1991. It was a simple language with basic data types, such as numbers and strings, and supported functions, modules, and exceptions. Over time, Guido and the Python community added many features to the language, such as support for object-oriented programming, improved exception handling, and a standard library of modules for common tasks.
- ▶ Python 2 was released in 2000, and was a significant improvement over the earlier versions. It introduced features such as list comprehensions, generators, and support for Unicode.
- ▶ Python 3 was released in 2008, and included many improvements and changes, such as better support for Unicode, improved handling of byte strings, and more consistent syntax.
- ▶ Today, Python is one of the most popular programming languages in the world, used by developers, scientists, researchers, and many others. It has a large and active community, which has contributed to the development of a vast array of libraries and frameworks that make Python even more versatile and powerful.

Guido Van Rossum



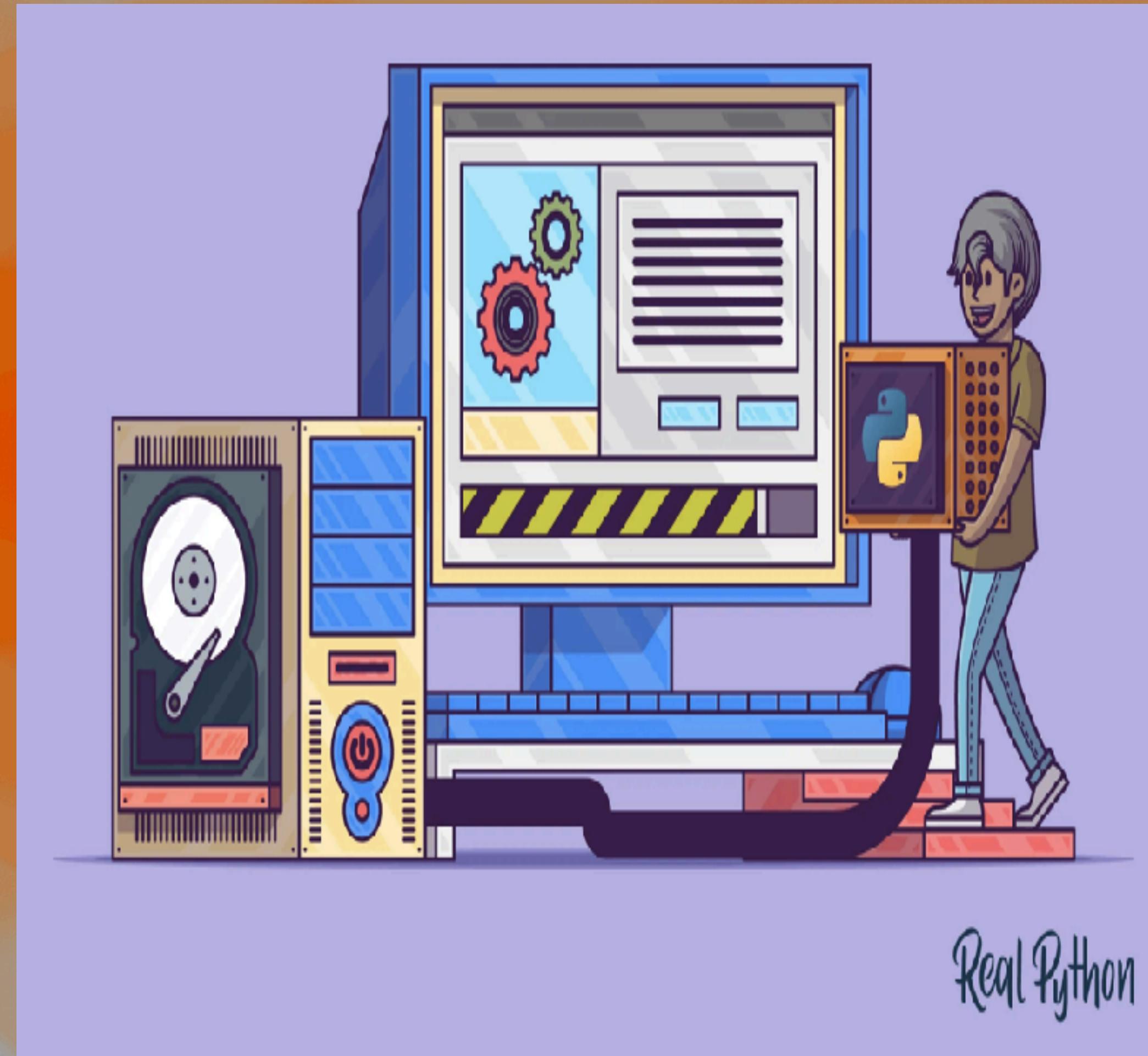
Named After This Show

Monty Python's Flying Circus



Setting Up Programming Environment

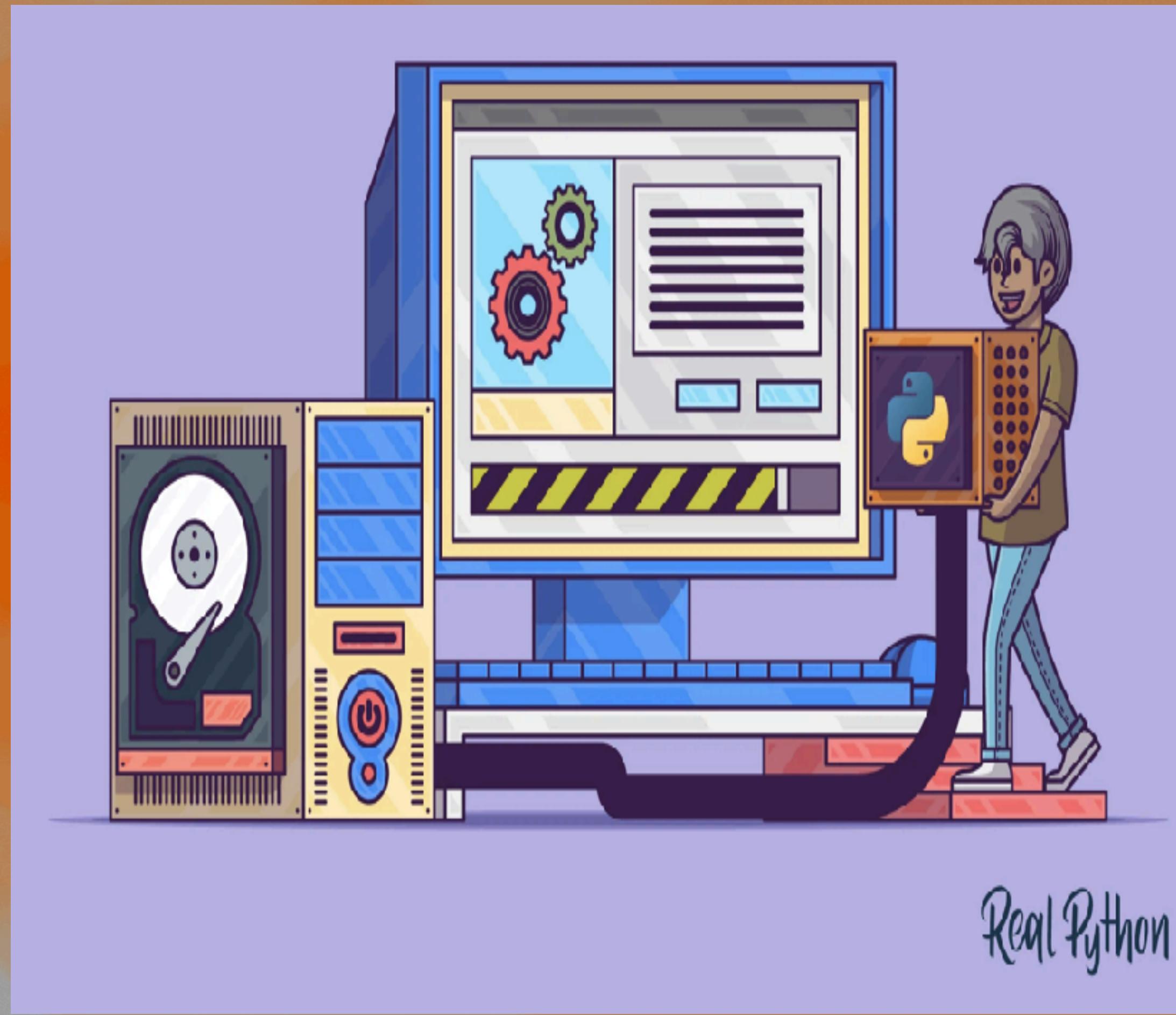
- ▶ **Download and Install Python:** Go to the official Python website and download the latest version of Python for Windows. Once the download is complete, run the installer and follow the prompts to install Python on your system.
- ▶ **Add Python to the PATH:** After installing Python, you need to add the Python executable to the PATH environment variable. This will allow you to run Python from any command prompt window without specifying the full path to the Python executable.
 - ▶ a. Open the Start menu and search for "Environment Variables".
 - ▶ b. Click on "Edit the system environment variables".
 - ▶ c. Click on the "Environment Variables" button.
 - ▶ d. Under "System Variables", scroll down and find the "Path" variable.



Real Python

Setting Up Programming Environment...

- ▶ e. Click on "Edit" and then "New".
- ▶ f. Add the path to your Python installation, which is usually "C:\PythonXX;C:\PythonXX\Scripts", where XX is the version number of Python you installed. Be sure to include the semicolon at the end of the path.
- ▶ Verify the installation: To verify that Python is installed correctly, open a new command prompt window and type "python" without the quotes. You should see the Python version number and the Python prompt ">>>".
- ▶ Install a code editor or IDE: While you can write Python code in any text editor, it is recommended to use a code editor or integrated development environment (IDE) that supports Python. Some popular options for Windows include Visual Studio Code, PyCharm, and IDLE (which comes bundled with Python).



Real Python

How Python Code is Being Executed?

- ▶ When you write a program in C/C++, you have to compile it. Compilation involves translating your human understandable code to machine understandable code, or Machine Code. Machine code is the base level form of instructions that can be directly executed by the CPU. Upon successful compilation, your code generates an executable file. Executing this file runs the operations in your code step by step.
- ▶ For the most part, Python is an interpreted language and not a compiled one, although compilation is a step. Python code, written in .py file is first compiled to what is called bytecode which is stored with a .pyc or .pyo format.
- ▶ Instead of translating source code to machine code like C++, Python code is translated to bytecode. This bytecode is a low-level set of instructions that can be executed by an interpreter. In most PCs, Python interpreter is installed at /usr/local/bin/python3.8. Instead of executing the instructions on CPU, bytecode instructions are executed on a Virtual Machine.

✿ ✿ ✿ Thank You ✿ ✿ ✿

Keep Smiling 😊