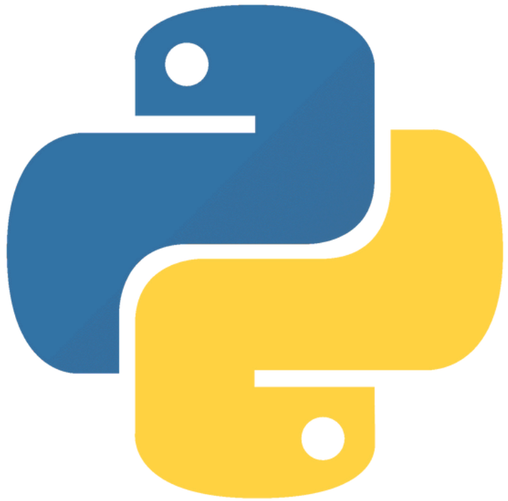
AI F5 SCHOOL

factoria F5
POWERED BY SIMPLON

Co-funded by
the European Union

Introducción a Python



python™

AI F5 SCHOOL

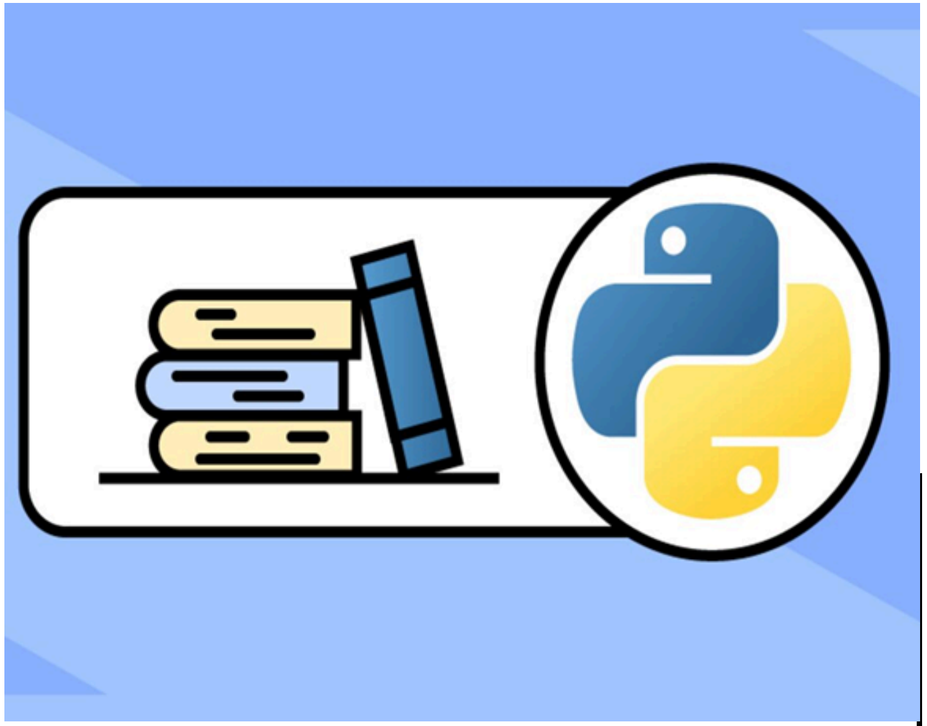
¿Que es Python?

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general que ha ganado una enorme popularidad desde su creación en 1991 por *Guido van Rossum*. Es conocido por su sintaxis clara y legible, y ademas Python soporta múltiples paradigmas de programación, incluidos el orientado a objetos, el imperativo, el funcional y el procedimental, lo que lo hace muy flexible y adaptable a diferentes tipos de proyectos.

Usos de Python

Python se utiliza en diversos campos gracias a su versatilidad:

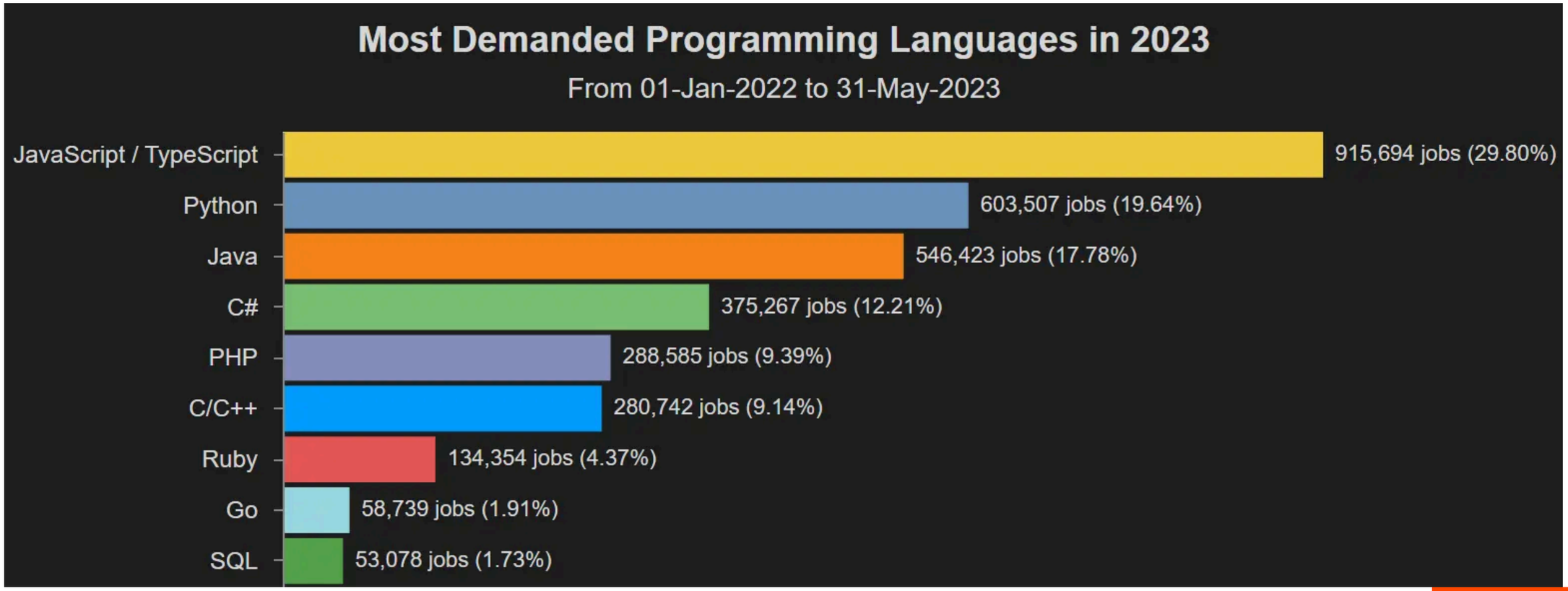
- Desarrollo Web con Django, Flask y fastAPI.
- Ciencia de Datos con NumPy, pandas, Matplotlib y Scikit-learn, etc.
- Automatización y Scripting para tareas repetitivas.
- Desarrollo de Software por su simplicidad y potencia.
- Juegos y Gráficos con Pygame.



Caracteristicas

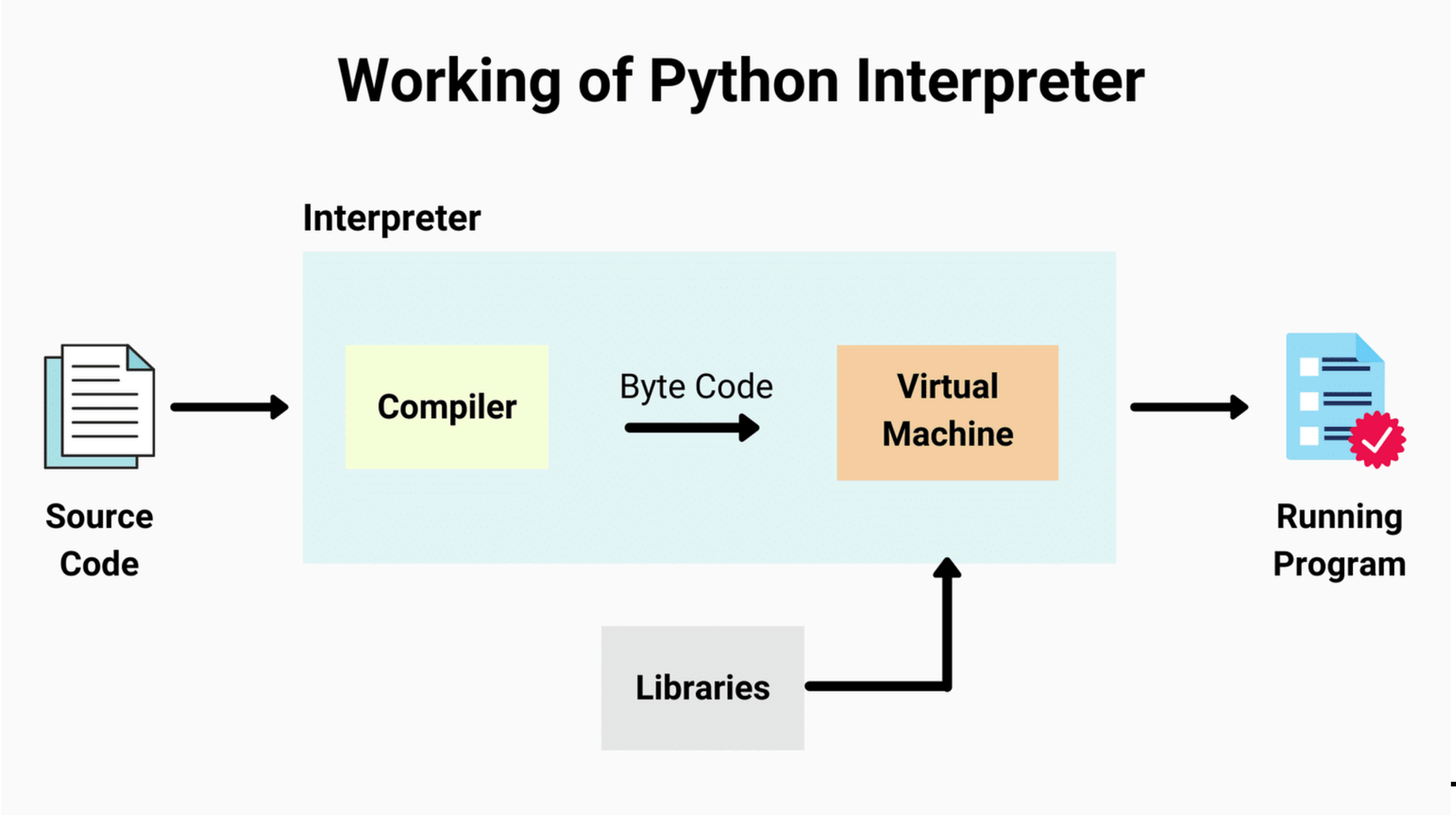
- Simple
- Fácil de aprender
- Libre de Código abierto
- Lenguaje de alto nivel
- Portable
- Interpretado
- Orientado a objetos
- Extensible
- Extensas bibliotecas

Actualidad de Python



Actualidad de Python

Interprete de Python



PIP

Instalador de dependencias
(Package Installer for Python)

Versión actual: 24

Comandos:

pip install
pip freeze

Convenciones Python

Guías de python

- PEP 8
- ZEN de Python (PEP 20)

```
>>> import this
"""
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
"""
```

Entornos virtuales

Cuando trabajas con Python, es una buena práctica utilizar entornos virtuales para gestionar tus dependencias de proyectos de manera aislada. Aquí tienes algunas opciones populares para crear y gestionar entornos virtuales en Python:

1. venv (módulo estándar de Python)

2. virtualenv

3. conda(Anaconda/Miniconda)

4. pipenv

5. poetry

Entornos de Desarrollo

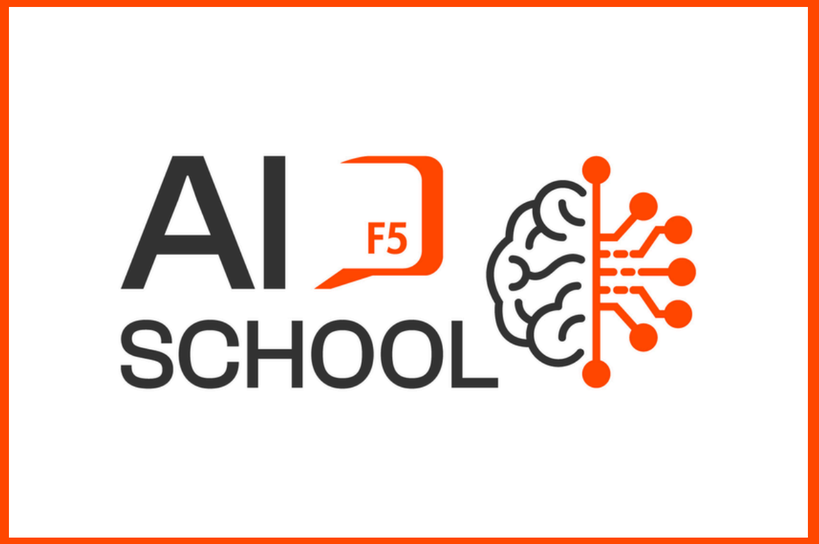
Entornos de desarrollo más usados en Python para ciencia de datos, incluyendo IDEs y notebooks:

1. Jupyter Notebooks
2. JupyterLab
3. PyCharm
4. Visual Studio Code (con extensiones para Python y Jupyter)
5. Spyder
6. Google Colab
7. Anaconda Navigator
8. Project IDX by Google

Jupyter NoteBook

Jupyter Notebook, anteriormente conocido como IPython Notebooks, es un entorno computacional interactivo basado en la web que permite crear documentos llamados notebooks. El término "notebook" puede referirse a varias cosas según el contexto, incluyendo la aplicación web de Jupyter, el servidor web Jupyter Python, o el formato de documento Jupyter. Un documento de Jupyter Notebook es un archivo JSON que sigue un esquema versionado y contiene una lista ordenada de celdas de entrada y salida. Estas celdas pueden incluir código, texto en Markdown, fórmulas matemáticas, gráficos y contenido enriquecido, y usualmente tienen la extensión ".ipynb".

Vamos al Notebook



F5
factoría
POWERED BY SIMPLON

Gracias