

Python para Análisis de Datos

Módulo 01

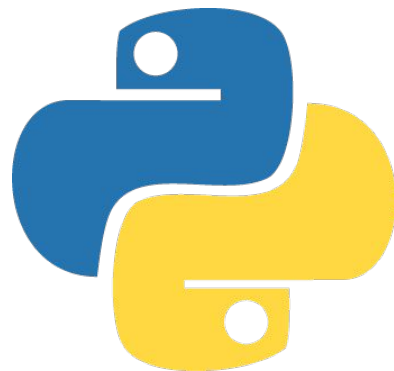
Introducción

Introducción

Python se ha convertido en el lenguaje de programación más utilizado a la hora de hacer análisis de datos. Sin embargo, no fue diseñado para ocupar este lugar, sino que evolucionó naturalmente en este rol.

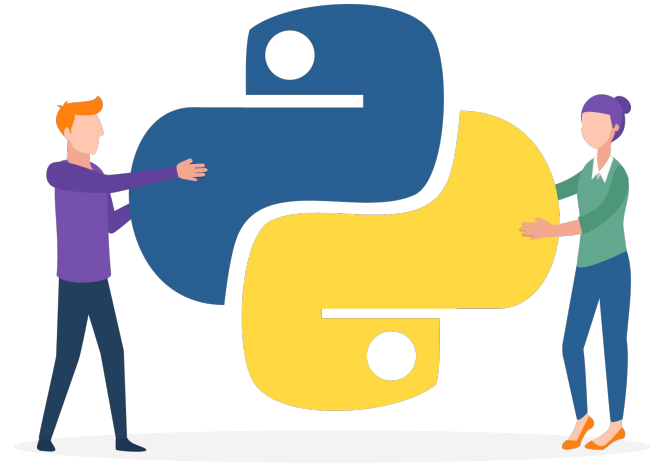
Fue la comunidad de usuarios la que adoptó al lenguaje y desarrolló el **ecosistema de librerías de análisis de datos que lo llevó a ser el más usado en ciencia de datos y machine learning.**

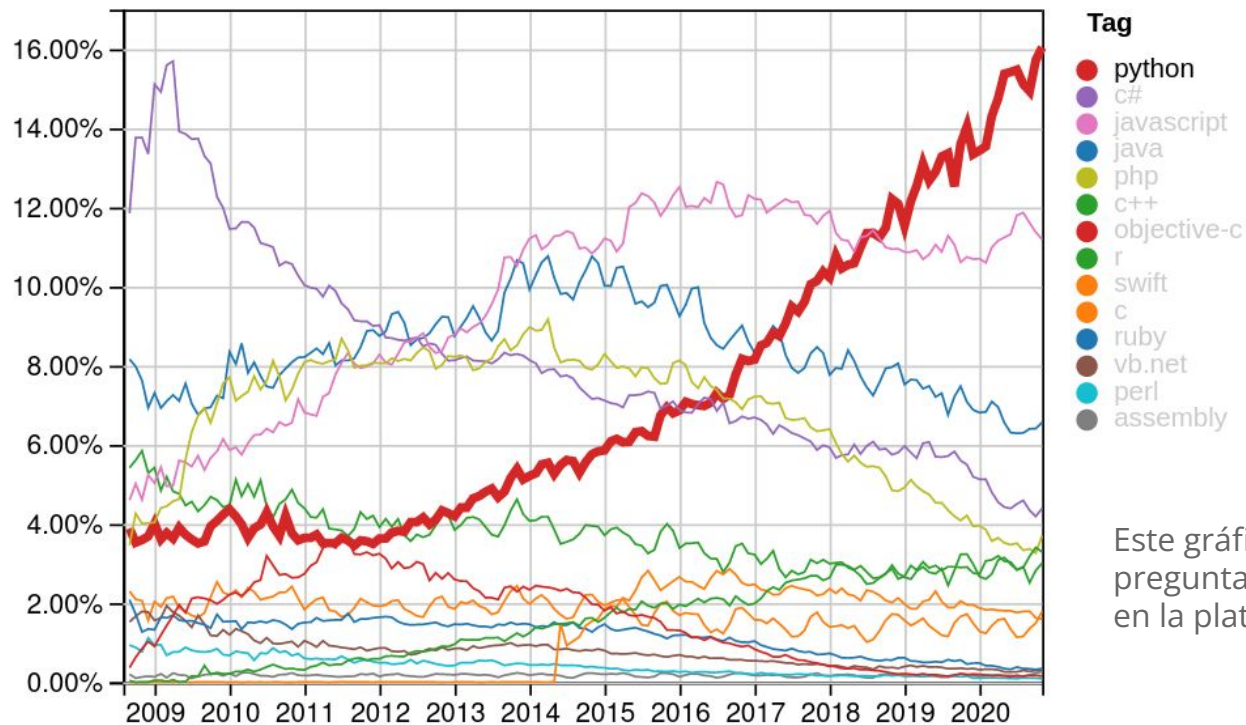
Sin embargo, hay que recordar que Python es un lenguaje de propósito general con numerosísimas aplicaciones, por lo que **está muy lejos de ser un lenguaje limitado al nicho del análisis de datos.**



El lenguaje se diseñó para ser fácil de entender y aprender, elegante y expresivo. Esto hizo que se convirtiera en la primera opción para mucha gente que se introduce en la programación y ha permitido que profesionales de las más variadas disciplinas elijan Python para llevar el análisis de datos al siguiente nivel en sus respectivos trabajos.

Según varias estadísticas, **es el lenguaje más aprendido de la actualidad.**





Este gráfico muestra el porcentaje de preguntas que se realizan por lenguaje en la plataforma Stackoverflow.

Librerías

Python tiene miles de librerías para las más variadas tareas, lo que permite realizarlas con relativa facilidad. En el entorno de análisis de datos, las tres más importantes son las que veremos en este curso:

Numpy, Pandas y Matplotlib.

- **Numpy** es una librería de cálculo numérico muy potente que forma la base de muchísimas otras librerías,
- **Pandas** está diseñada para el análisis y manejo de datos en estructuras tabulares, y
- **Matplotlib** es la base para realizar gráficos en Python.



Anaconda

Python es open source y puede ser usado y distribuido libremente (otra de sus grandes ventajas). Es por eso que se ha creado una **distribución particular pensada para ciencia de datos y machine learning: Anaconda.**

Esta distribución viene con las librerías más importantes para análisis de datos, junto con un instalador de paquetes (pensado para manejar las dependencias entre estos paquetes de manera más efectiva), una interfaz gráfica y algunas aplicaciones como Jupyter Notebook y Spyder.

La versión individual se puede descargar de esta página:
<https://www.anaconda.com/products/individual>



Jupyter Notebook

Es un entorno interactivo para trabajar con lenguajes de programación. Es una **aplicación web**, por lo que se visualiza en cualquier navegador.

Está pensado para articular código ejecutable, gráficos, texto, tablas, fórmulas, etc. en un solo documento.

Esto hace que sea muy cómodo ver visualizaciones mientras se exploran los datos y que sea posible **usar la misma herramienta tanto para el desarrollo como para la presentación.**



El documento está estructurado en celdas que pueden ser de código o de markdown.

Las celdas de código pueden ejecutarse independientemente y el orden en el que fueron ejecutadas se muestra al lado del In y el resultado de la última expresión, si lo hay, se marca con Out. Sin embargo, todo pertenece a la misma sesión y las variables creadas en celdas anteriores permanecen en la memoria.

```
In [1]: 1 print("Hola mundo")
        Hola mundo

In [2]: 1 2+2
Out[2]: 4
```



Markdown

Las celdas también pueden ser de tipo **markdown**, un lenguaje de marcado usado para formatear texto con un código muy simple. Además, soporta código html.

```
1 # Título 1
2
3 ## Título 2
4
5 ### Título 3
6
7 *itálicas*
8 **negrita**
9 `monospace`
10
11 - lista
12 - lista
13
14
15 1. punto 1
16 2. punto 2
17
18 Texto principal
19
20 Cita
21
```



Título 1

Título 2

Título 3

itálicas

negrita

`monospace`

- lista
- lista

1. punto 1
2. punto 2

Texto principal

Cita

¡Muchas gracias!

¡Sigamos trabajando!