

INTRO A PYTHON PARA CIENCIA DE DATOS



Clase 1:

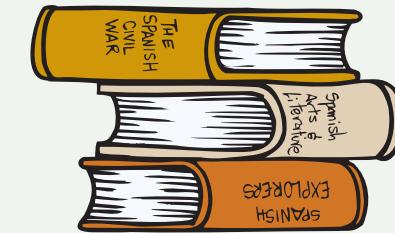
Introducción

20/04/2024



- ¿Por qué Python?
- Entornos de desarrollo (IDE)
- Fundamentos de la programación
- Tipos de datos
- Operaciones y condicionales
- Introducción a funciones

EL CURSO



- **Clases:** Teórico prácticas que quedan grabadas.
- **Material:** Repositorio de Github
- **Consultas Python:** Comunidad LVLUP en Discord.
Canal del curso #intro-python-ciencia-de-datos
(link)
- **Consultas Administrativas:** Grupo Whatsapp del
Curso.

¿POR QUÉ PYTHON?

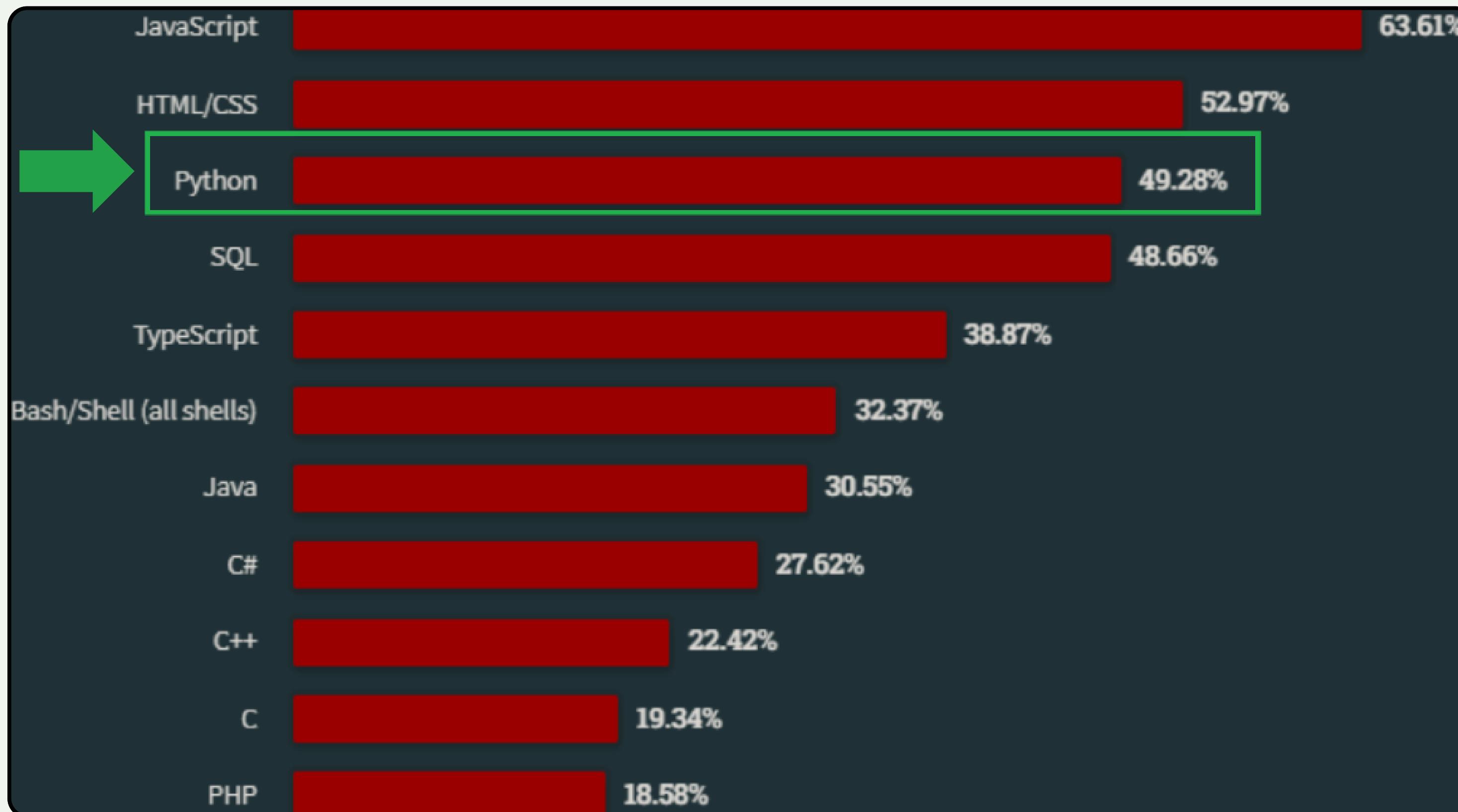
CONTEXTO



- Creado por el holandés **Guido Van Rossum** en 1990
- Su **objetivo** era un lenguaje que fuera:
 - Simple
 - Alto Nivel (legible)
 - Versátil y flexible
 - Código abierto (Open Source)
- “**The Zen of Python**”:
 - Bello es mejor que feo
 - Simple es mejor que complejo
 - Explícito es mejor que implícito

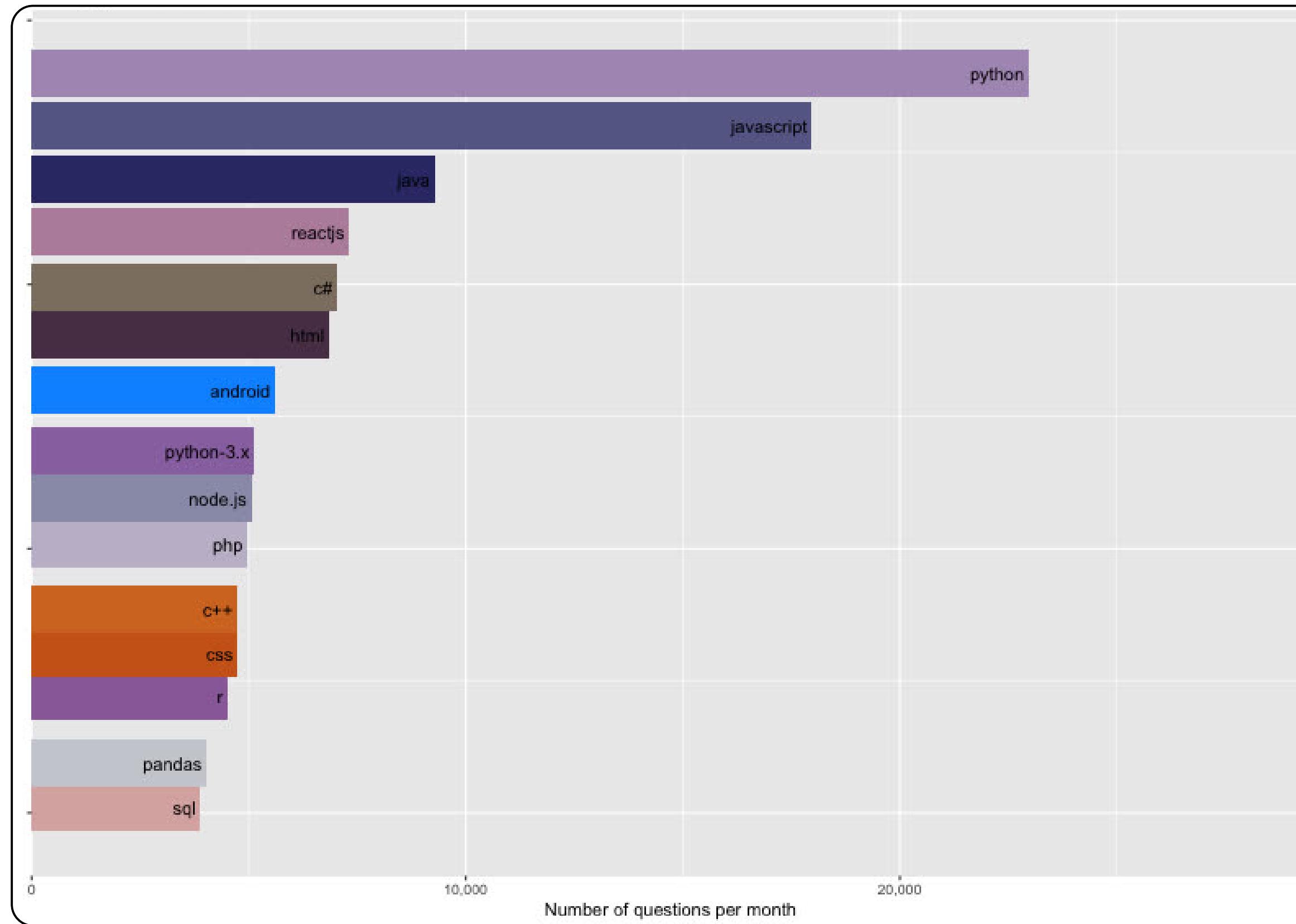
STACK OVERFLOW LENGUAJES MÁS POPULARES 2023

LevelUP



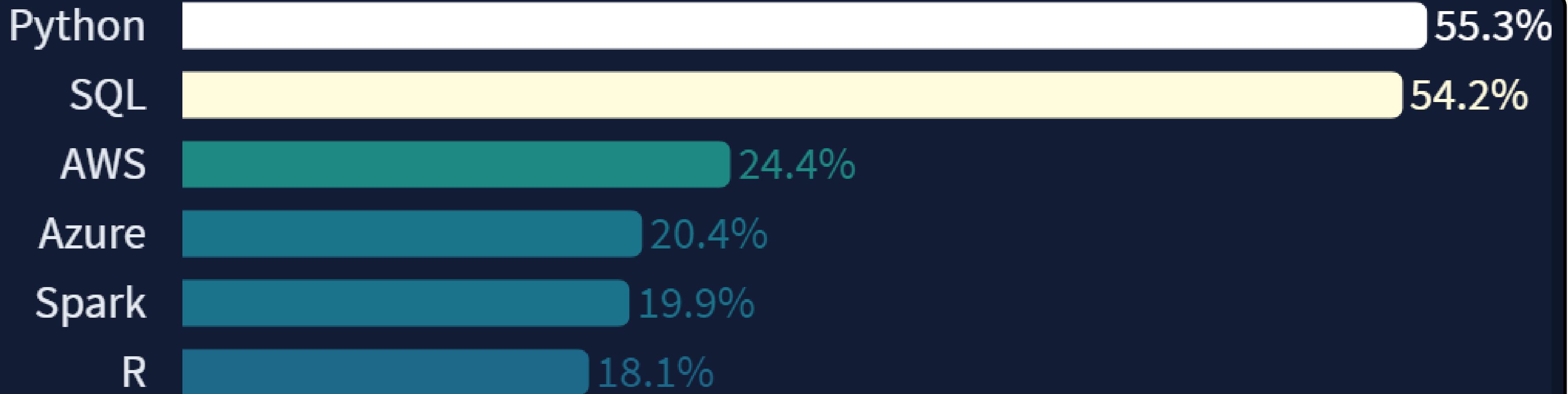
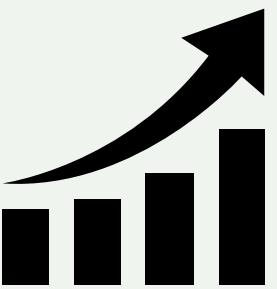
TOP STACK OVERFLOW QUESTIONS

LevelUP

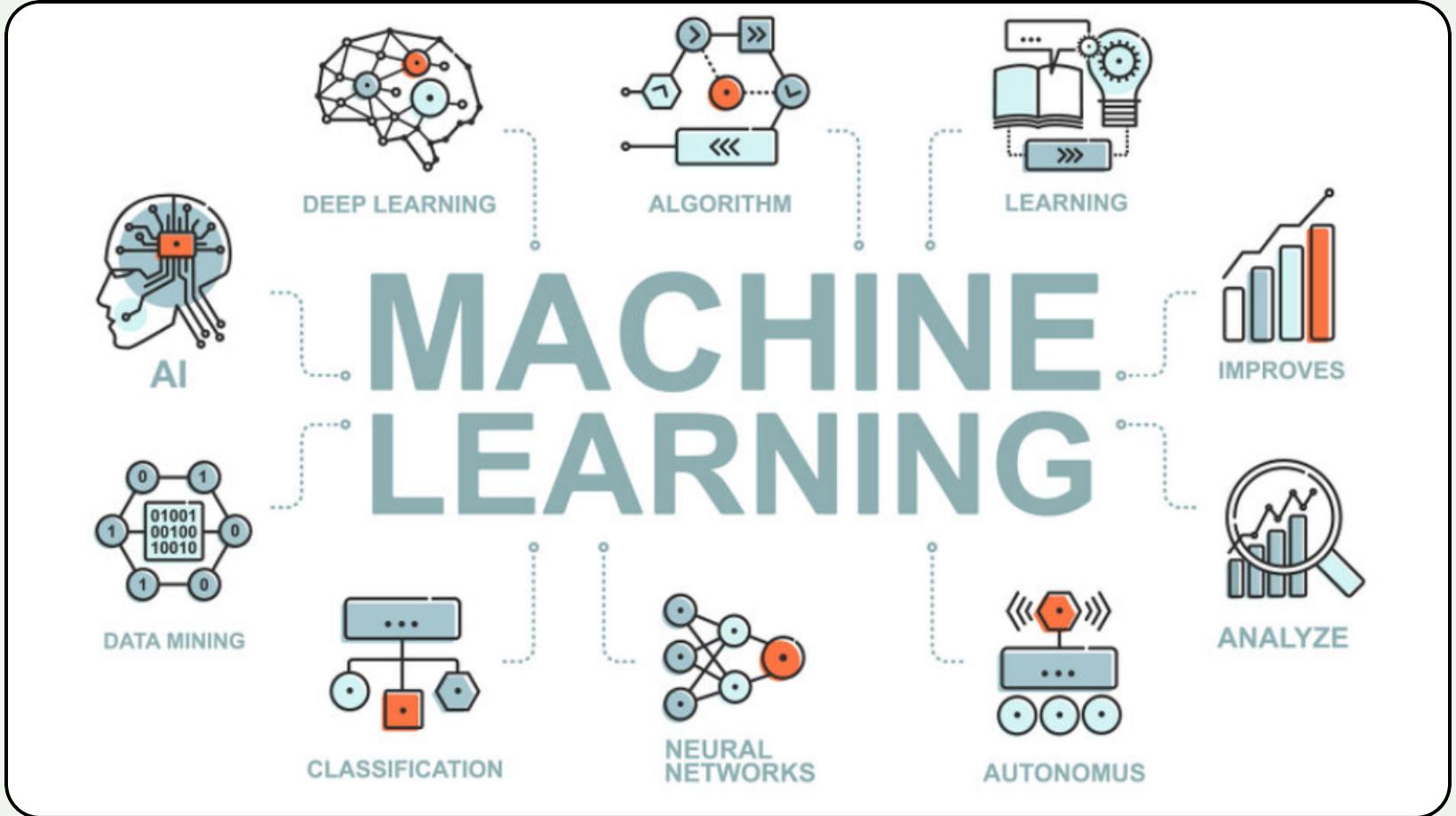


<https://stackoverflow.blog/2023/01/26/comparing-tag-trends-with-our-most-loved-programming-languages/>

TOP DATA ANALISYS SKILLS



¿DÓNDE SE USA PYTHON?



CAMPOS DE APLICACION

- Ingeniería
- Entretenimiento
- Medicina
- Psicología
- Finanzas
- Otros

PAPERS



John Loewen, PhD in Towards AI

The Art of Prompting GPT-4: Python CSV Cleaning and Data Visual Code

Simple creation of Python scripts for interactive choropleth maps

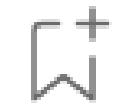
★ · 7 min read · 5 days ago



245



2



...



<https://medium.com/towards-artificial-intelligence/the-art-of-prompting-gpt-4-python-csv-cleaning-and-data-visual-code-2d9b8ca6b6f6>

PAPERS



JMIR Medical Informatics
Clinical informatics, decision support for health professionals, electronic health records, and ehealth infrastructures.
ISSN 2291-9694

About Search Archive Current Issue Submit Editorial Board

[JMIR Med Inform.](#) 2019 Jan-Mar; 7(1): e12414.

PMCID: PMC6371067

Published online 2019 Jan 29. doi: [10.2196/12414](https://doi.org/10.2196/12414)

PMID: [30694199](#)

Data Analysis and Visualization of Newspaper Articles on Thirdhand Smoke: A Topic Modeling Approach

Monitoring Editor: Gunther Eysenbach

Reviewed by Casper Zhang and Anuja Majmundar

[Qian Liu](#), BTech, MSc, PhD,^{1,2} [Qiuyi Chen](#), BA,¹ [Jiayi Shen](#), MBBS,³ [Huailiang Wu](#), MBBS,³ [Yimeng Sun](#), BA,¹ and [Wai-Kit Ming](#), MD, MPH, MMSCI, PhD^{4,5,6,7}

The Python Papers

[Subscribe](#)[Sign in](#)

How to Send Emails with Python



MIKE DRISCOLL

SEP 18, 2023 · PAID

[Share](#)

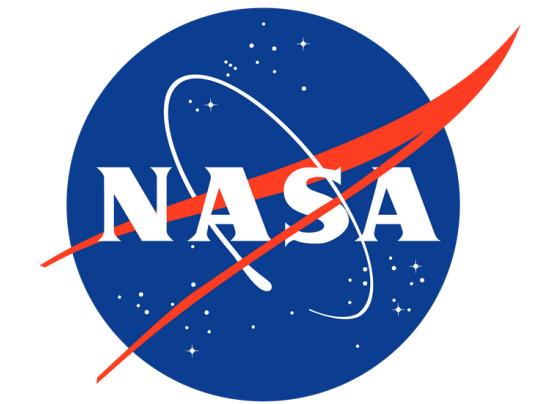
Python provides a couple of really nice modules that you can use to craft emails with. They are the `email` and `smtplib` modules. Instead of going over various methods in these two modules, you'll spend some time learning how to actually use these modules.

EMPRESAS/APLICACIONES QUE LO USAN

Google



NETFLIX



BATTLEFIELD™



amazon

ENTORNOS DE DESARROLLO (IDE)



IDE (INTEGRATED DEVELOPMENT ENVIRONMENT)

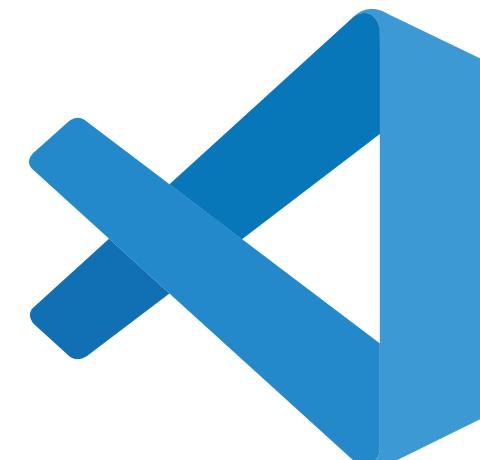
Permiten:

✍ Escribir código

✓ Compilar

👾 Depurarlo

✓ Ejecutar



¿POR QUE COLAB?

Plataforma *gratuita* de Google que permite escribir y ejecutar código Python en un entorno basado en la nube, sin necesidad de configurar un entorno de desarrollo local (sin instalar nada en nuestra PC). Proporciona acceso a recursos informáticos especialmente para tareas de **Ciencia de Datos y Machine Learning**.

 Algunas características:

 Ejecución interactiva

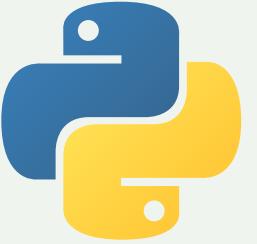
 Integración con otras herramientas como google sheets.

 Storytelling

FUNDAMENTOS DE LA PROGRAMACION



¿Qué es Programar ? ¿Qué es un Flujo?



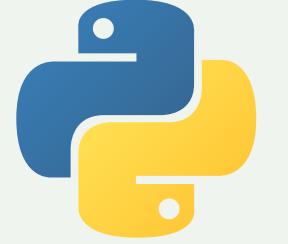
👉 Es indicarle a la computadora los pasos a seguir (*instrucciones*) para realizar una tarea.

```
 62 self.file = None
 63 self.fingerprints = set()
 64 self.logdups = True
 65 self.debug = debug
 66 self.logger = logging.getLogger(__name__)
 67 if path:
 68     self.file = open(os.path.join(path, 'fingerprint.log'), 'w')
 69     self.file.seek(0)
 70     self.fingerprints.update(self.file.readlines())
 71
 72 @classmethod
 73 def from_settings(cls, settings):
 74     debug = settings.getboolean('FINGERPRINT_DEBUG')
 75     return cls(job_dir(settings), debug)
 76
 77 def request_seen(self, request):
 78     fp = self.request_fingerprint(request)
 79     if fp in self.fingerprints:
 80         return True
 81     self.fingerprints.add(fp)
 82     if self.file:
 83         self.file.write(fp + os.linesep)
```

👉 Conjunto de *instrucciones* que se llevan a cabo durante la ejecución de un programa.
Es posible ***controlar o modificar*** el flujo para que tome ciertas acciones, bifurcaciones o repeticiones según lo hayamos definido.

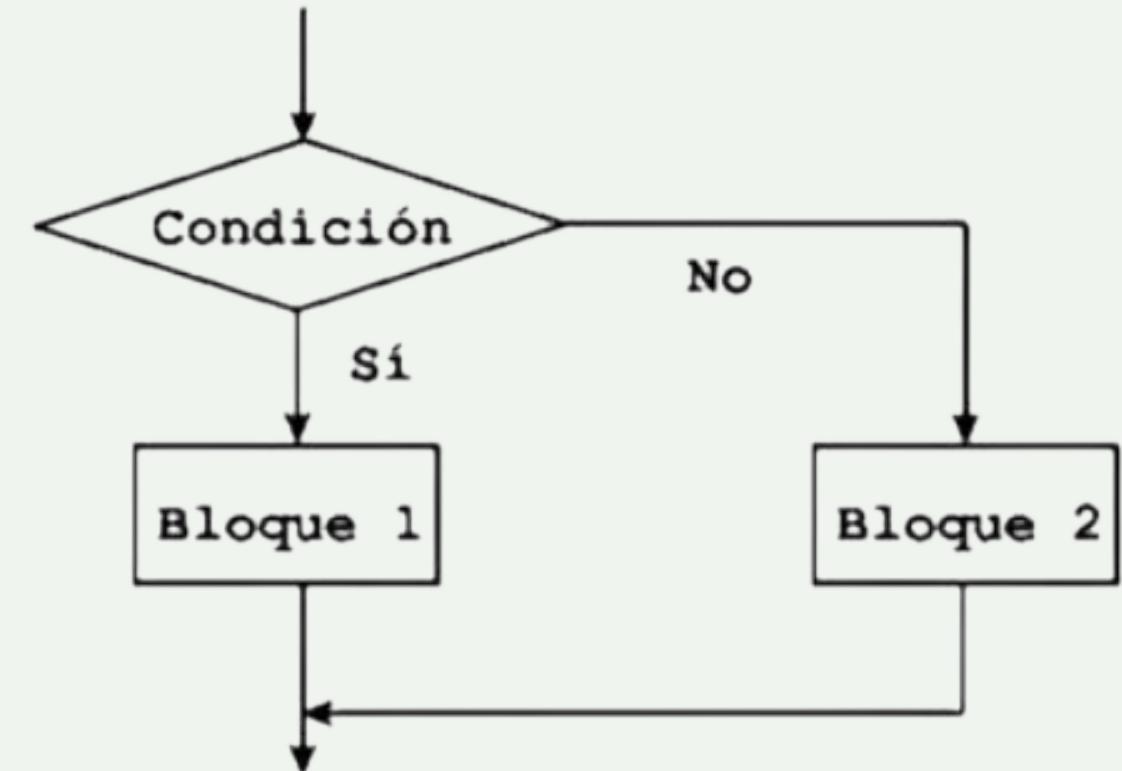


Algoritmos y Estructuras de control



Los **algoritmos** son secuencias lógica de pasos o instrucciones para definir el flujo a realizar para llevar a cabo una tarea.

A través de las **estructuras de control**, podemos controlar dicho flujo (**if-else, for-while, break-continue**)

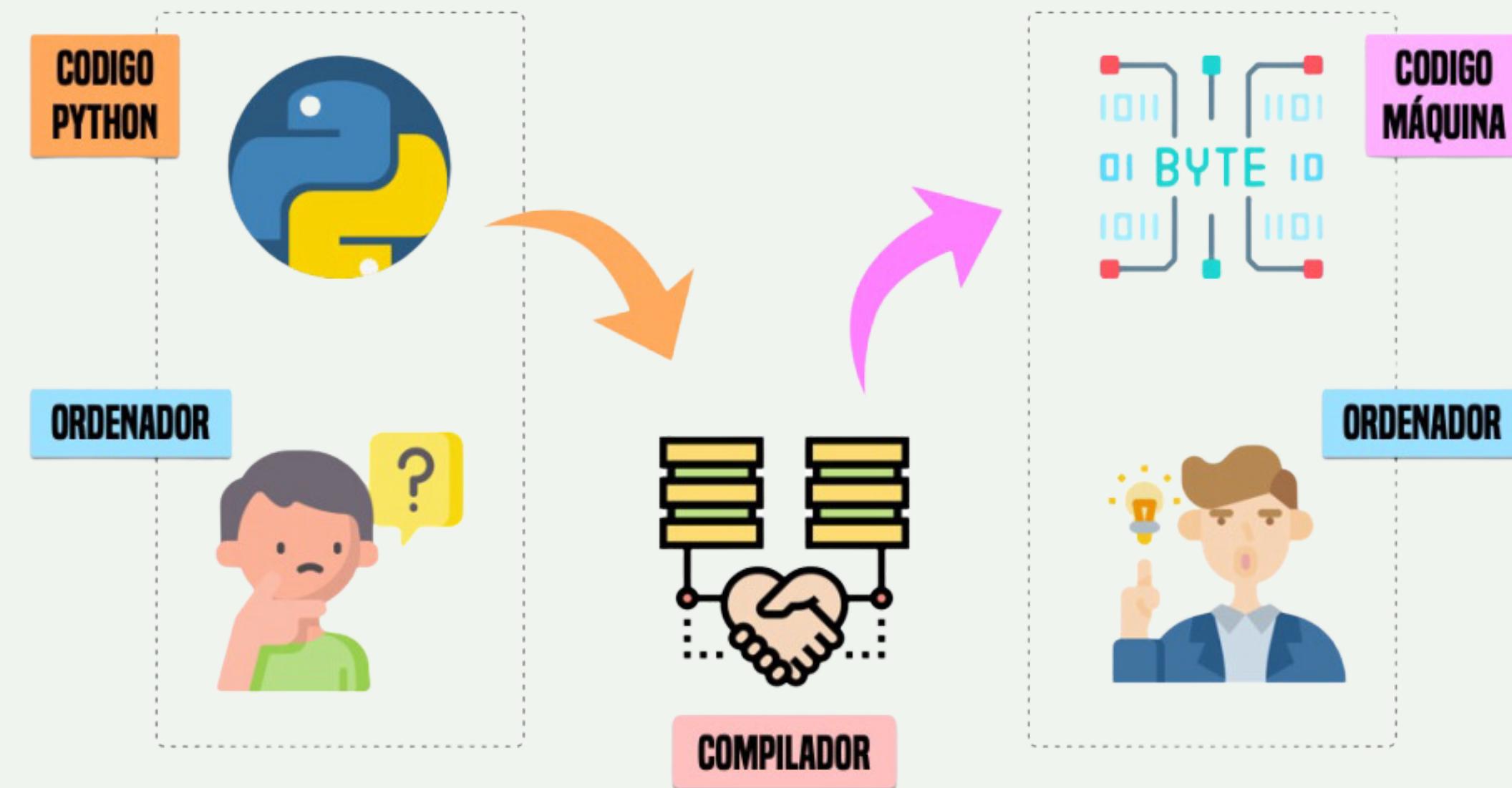


```
if condición  
    bloque-1  
else  
    bloque-2  
end
```

...

¿Cómo se ejecuta? Compilado & Interpretado

Los compiladores son programas que convierten tu código en código máquina (**ceros y unos**)



¿Cómo se ejecuta? Compilado vs Interpretado

En Python el código primero se **compila** a ByteCode y luego ejecuta el **intérprete** (No hay compilación directa a código máquina)

✓ Se va ejecutando línea por línea facilitando el *debuggin*

CÓDIGO FUENTE → INTÉRPRETE

```
import this  
print("Hello, World!")
```



LINEA POR LINEA
→

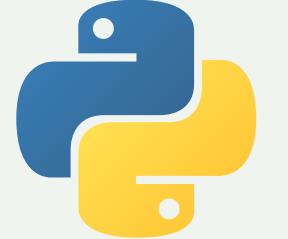
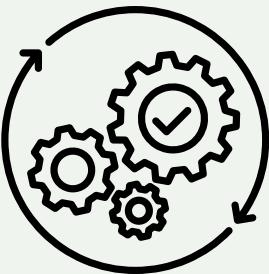
BYTE CODE

→ SALIDA

```
0 LOAD_GLOBAL  
0 <print>  
1 <'Hello, World!'>  
2 LOAD_CONST  
4 CALL_FUNCTION 1
```



¿Cuáles son las mejores prácticas ?



✓ PEP 8 (Python Enhancement Proposal)

"Imports should be grouped in the following order:

1. Standard library imports.
2. Related third party imports.
3. Local application/library specific imports."

Correct:
spam(1)

Wrong:
spam (1)

Correct:
x = 1
y = 2
long_variable = 3

Wrong:
x = 1
y = 2
long_variable = 3

"""Return a foobang

Optional plotz says to frobnicate the bizbaz first.
"""

TIPOS DE DATOS Y VARIABLES



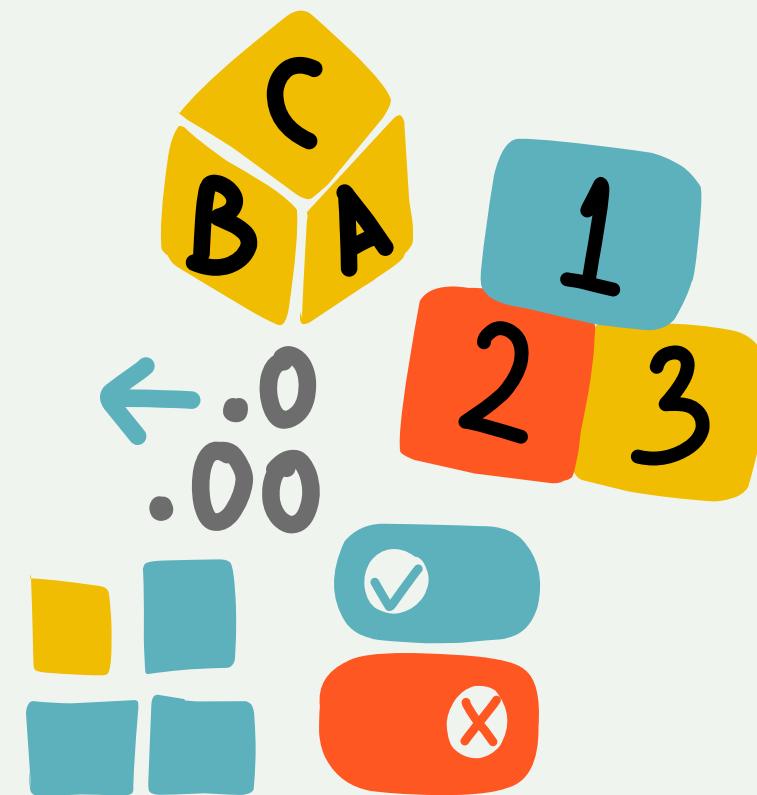
Tipos de datos



Es la clasificación de los valores que pueden ser representados y manipulados por un programa, determinando las operaciones que se pueden realizar sobre ellos.

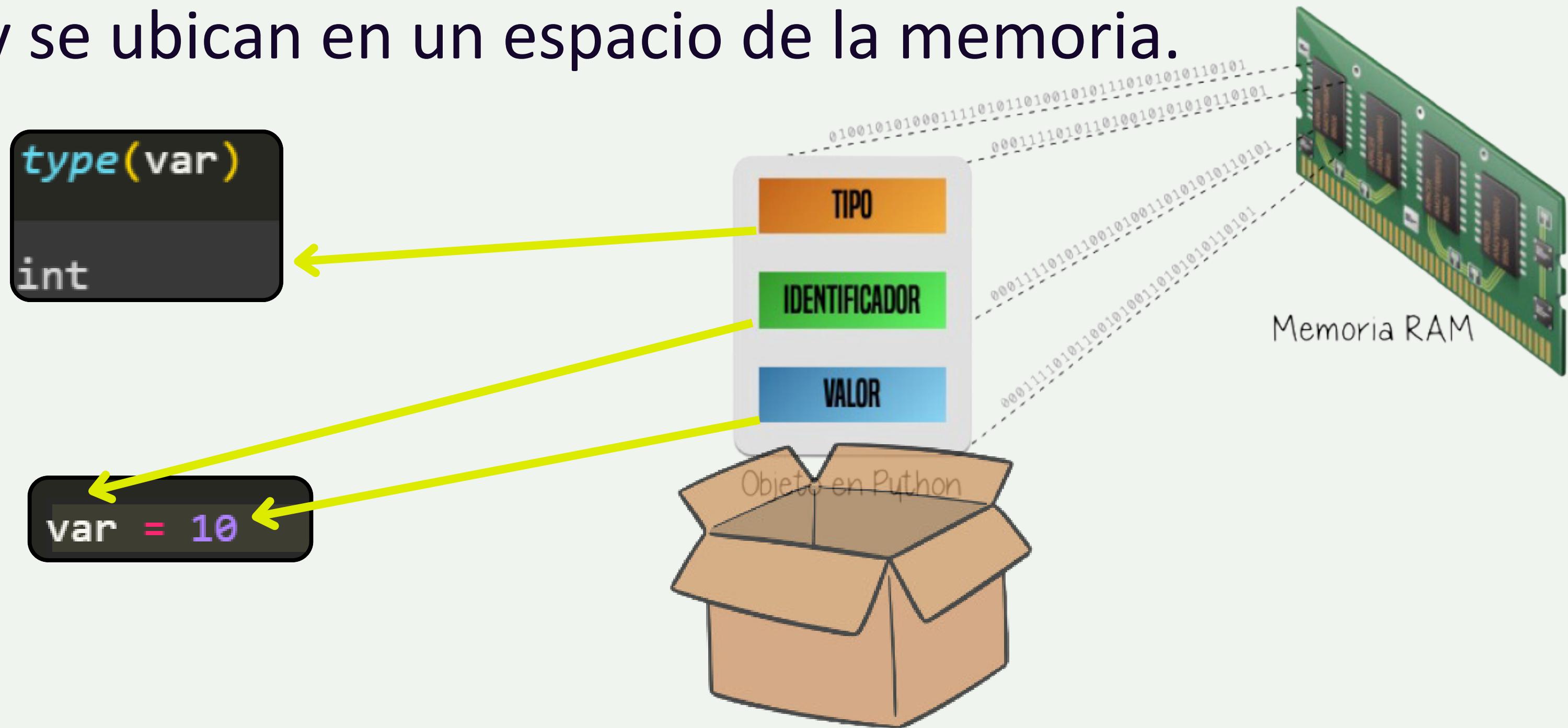
Tipos básicos:

- **String:** “Hola LVL UP”
- **Int:** 10
- **Float:** 3.14
- **Bool:** True/False
- **Date:** 20/04/2024

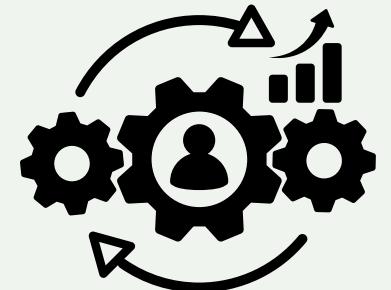


Variables

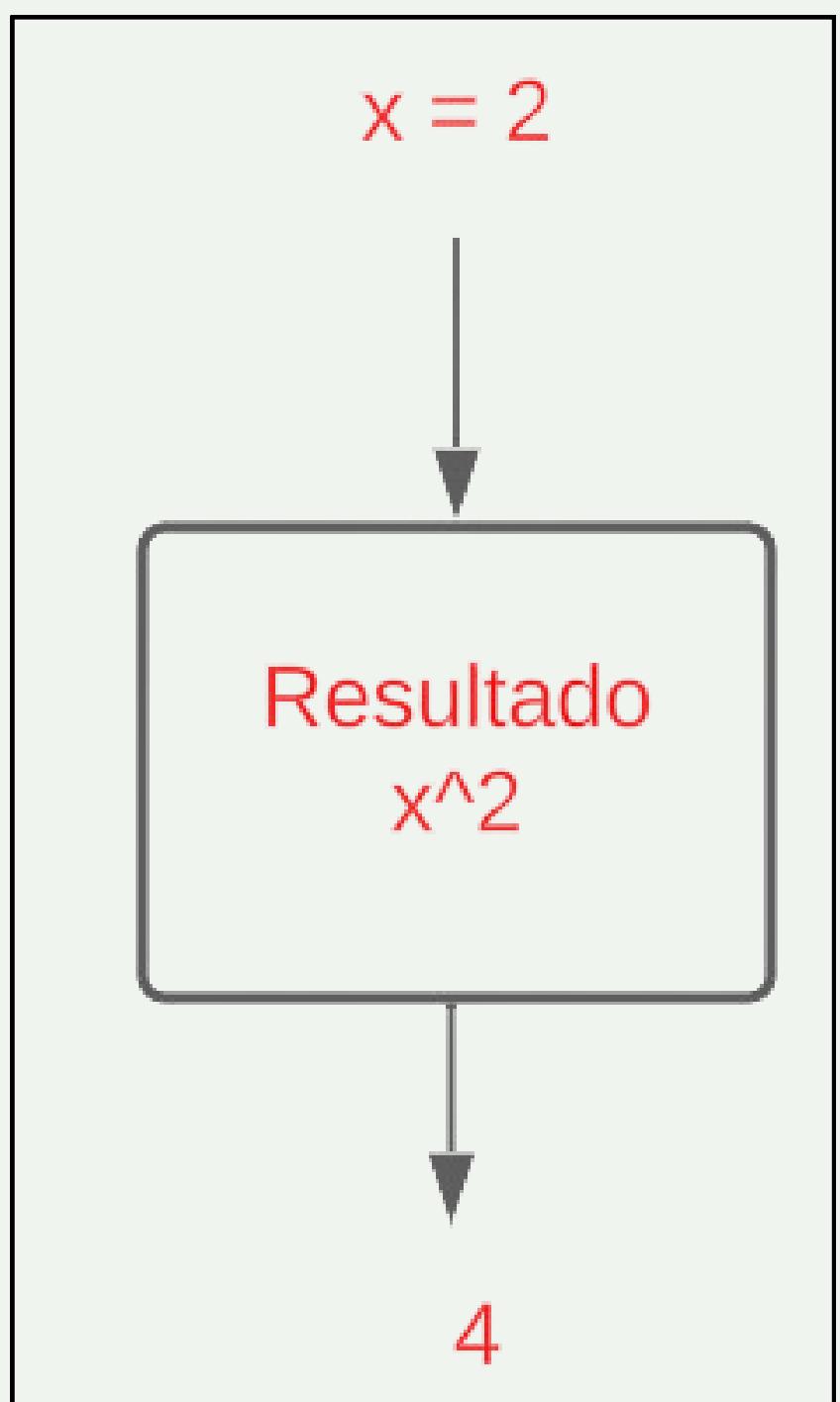
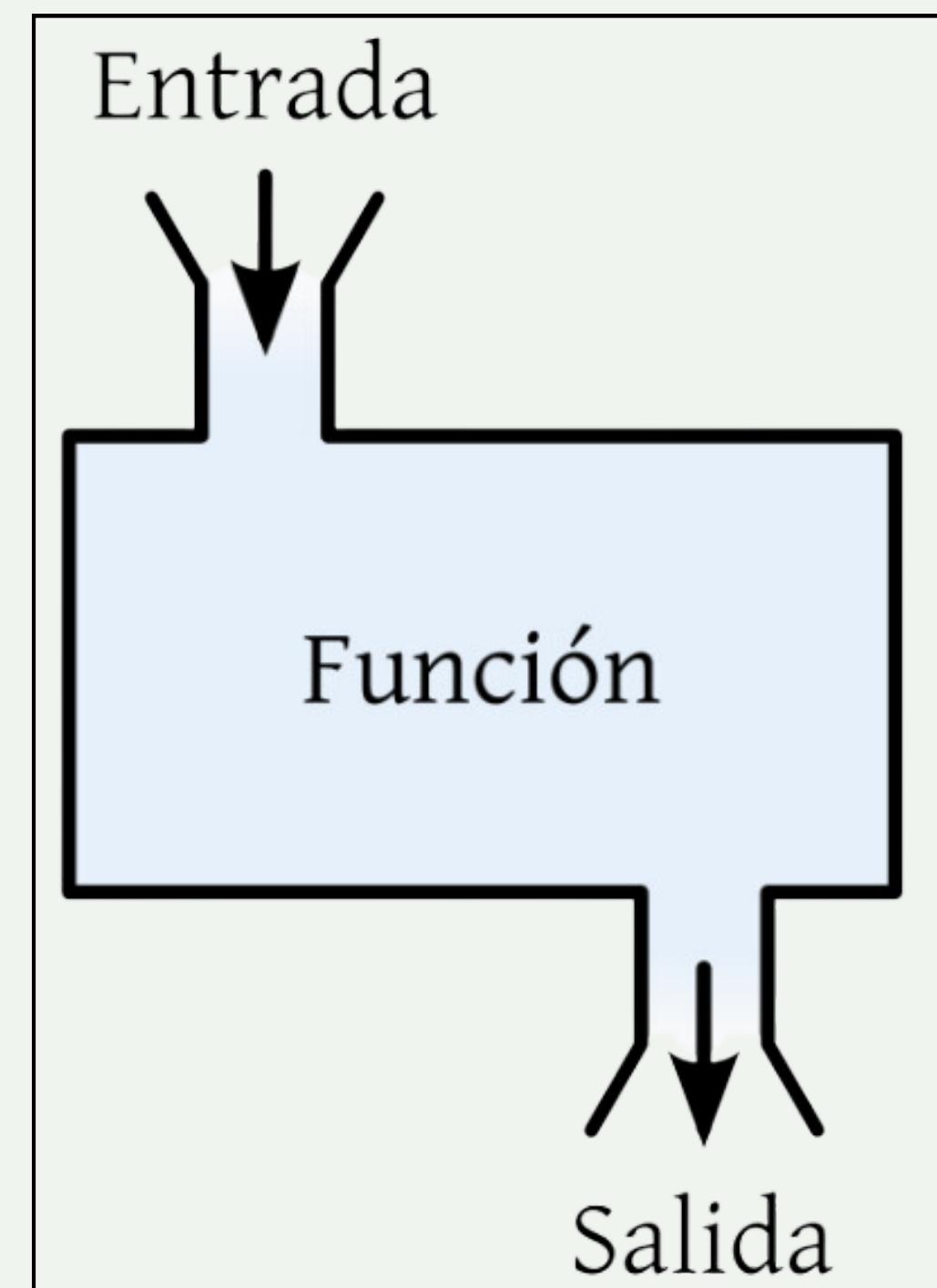
Las *variables* son “contenedores” o “cajas” que guardan un valor o dato al cual referencian (o como veremos en el curso “*objetos*”) y se ubican en un espacio de la memoria.



Funciones



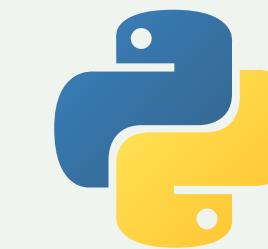
Como en matemática, una función es una “*máquina*” que recibe una entrada (datos, información) y devuelve una salida o resultado.



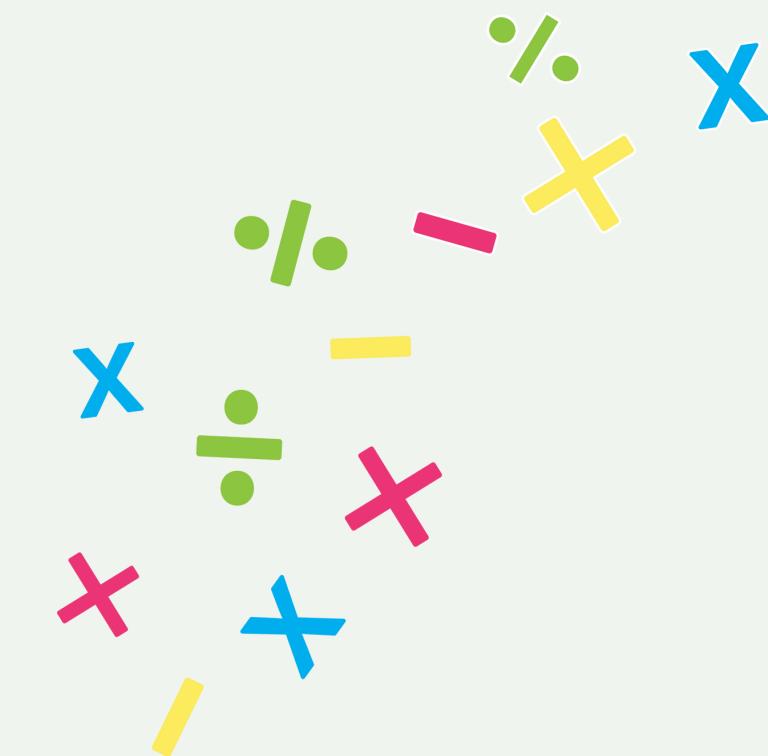
OPERACIONES Y CONDICIONALES



Operaciones



- ARITMÉTICAS
- ASIGNACIÓN
- COMPARACIÓN
- LÓGICAS
- IDENTIDAD
- PERTENENCIA

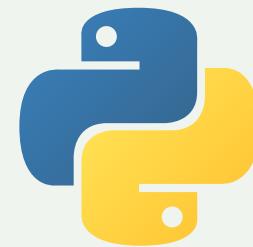


p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F



nombre  valor

Condicionales



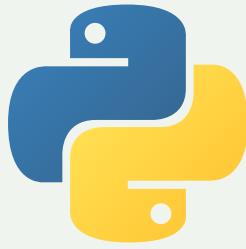
Permiten tomar **decisiones** en el flujo en base a el cumplimiento o no de una determinada condición.

El más conocido es el **if/elif/else**, pero también existen otros como **match/case** que otorgan una funcionalidad similar.

VAMOS AL COLAB...



Workshop



A thumbnail image for a workshop titled "Tus primeros pasos en Python para Análisis de Datos". The thumbnail features the LevelUP logo, the Python logo, and a large red play button. The background shows a close-up of a snake's head against a dark, futuristic interface with multiple screens displaying data.

WORKSHOP Tus primeros pasos en Python

Workshop.

Tus primeros pasos en Python

para Análisis de Datos

Watch on YouTube