

Workshop:



Tus primeros pasos en Python

Acelerador de Carrera



Temario

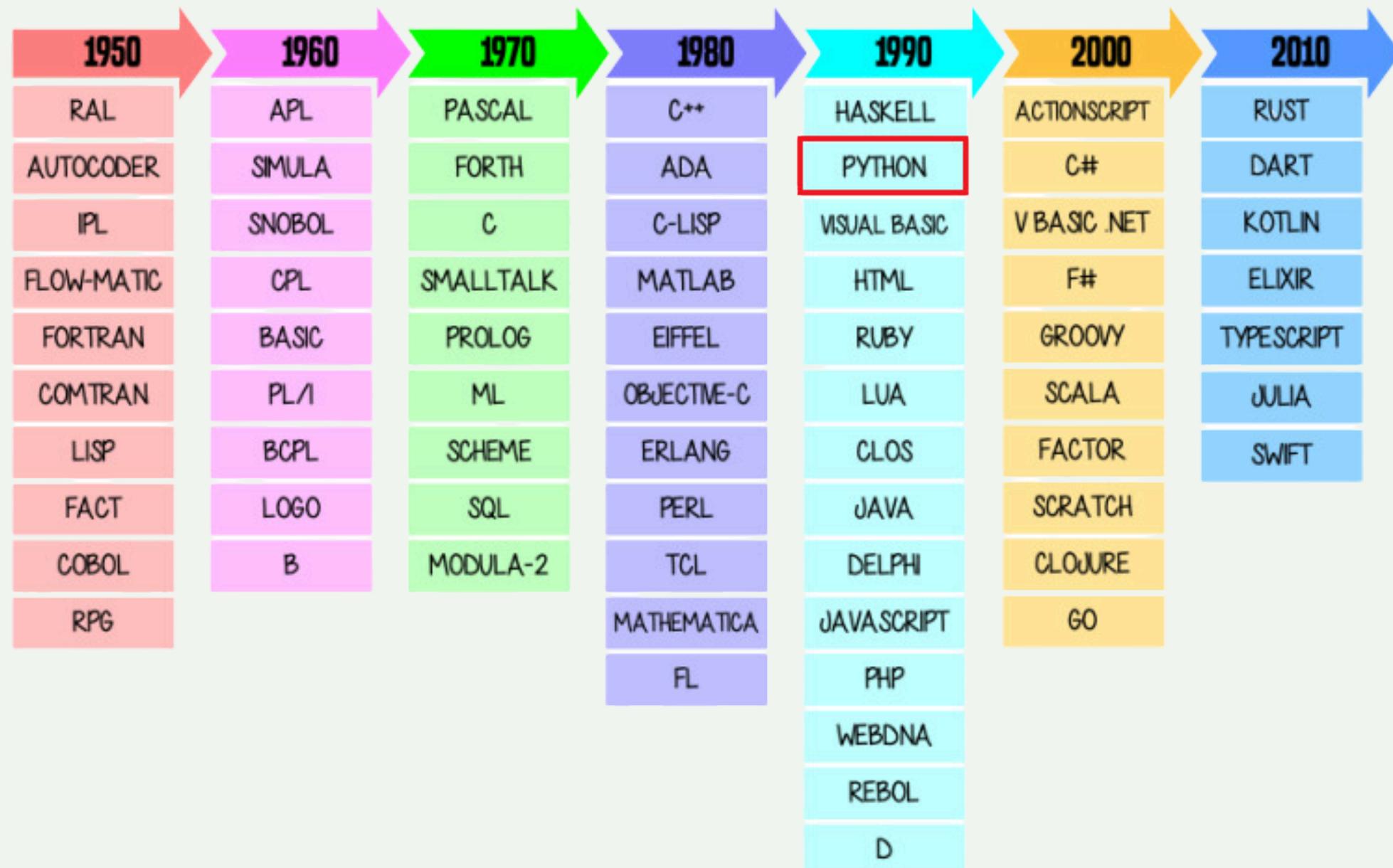
- ¿Por qué Python?
- ¿Dónde se usa?
- Conceptos Fundamentales

¿POR QUÉ PYTHON?

UN POCO DE HISTORIA



Creado por el holandés *Guido Van Rossum* en 1990



UN POCO DE HISTORIA

Su objetivo era un lenguaje que fuera:

- Simple
- Alto Nivel (legible)
- Versátil y flexible
- Código abierto (Open Source)



THE ZEN OF PYTHON

```
1 Beautiful is better than ugly.
2 Explicit is better than implicit.
3 Simple is better than complex.
4 Complex is better than complicated.
5 Flat is better than nested.
6 Sparse is better than dense.
7 Readability counts.
8 Special cases aren't special enough to break the rules.
9 Although practicality beats purity.
10 raise PythonicError("Errors should never pass silently.")
11 # Unless explicitly silenced.
12 In the face of ambiguity, refuse the temptation to guess.
13 There should be one-- and preferably only one --obvious way to do it.
14 # Although that way may not be obvious at first unless you're Dutch.
15 Now is better than ...               never.
16 Although never is often better than rightnow.
17 If the implementation is hard to explain, it's a bad idea.
18 If the implementation is easy to explain, it may be a good idea.
19 Namespaces are one honking great idea -- let's do more of those!
```

- **Bello** es mejor que feo.
- **Explícito** es mejor que implícito.
- La **legibilidad** es importante
- **Espaciado** es mejor que denso.
- Evitar ambigüedades.
- **Simple** es mejor que **complejo**.

SINTAXIS: PYTHON VS C++

“La sintaxis para imprimir *un mensaje* en
C++ no puede ser tan difícil...”

SINTAXIS: PYTHON VS C++

La sintaxis

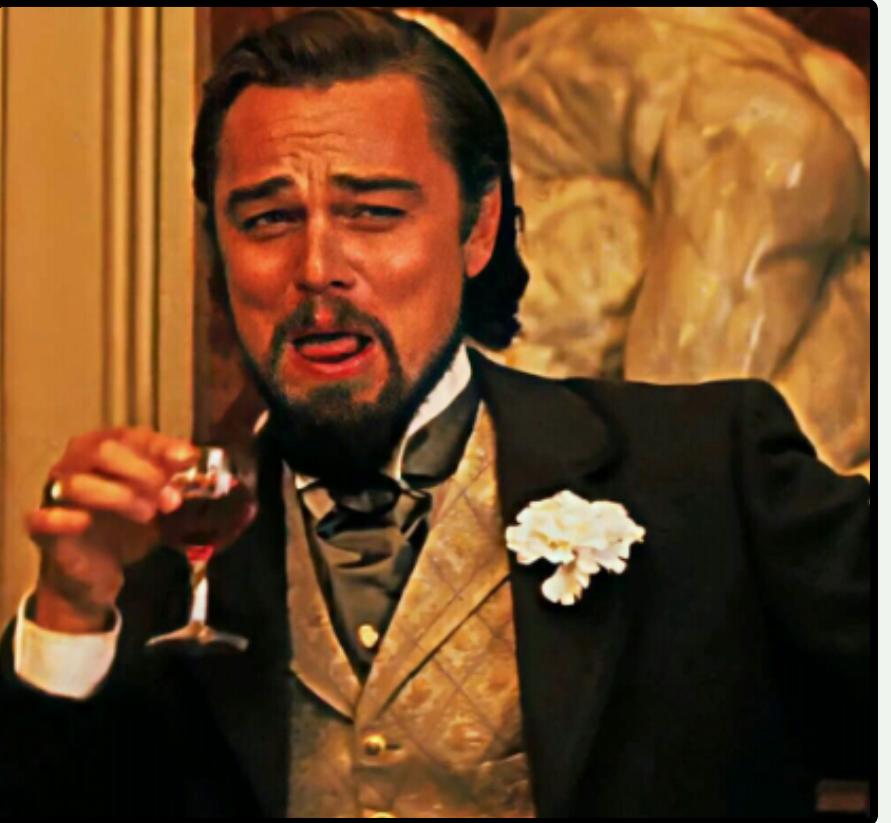
```
#include <iostream>

int main() {
    std::cout << "Hola comunidad LVL-UP" << std::endl;
    return 0;
}
```



SINTAXIS: PYTHON VS C++

Ah pero en *Python*...



```
print("Hola comunidad LVL-UP")
```

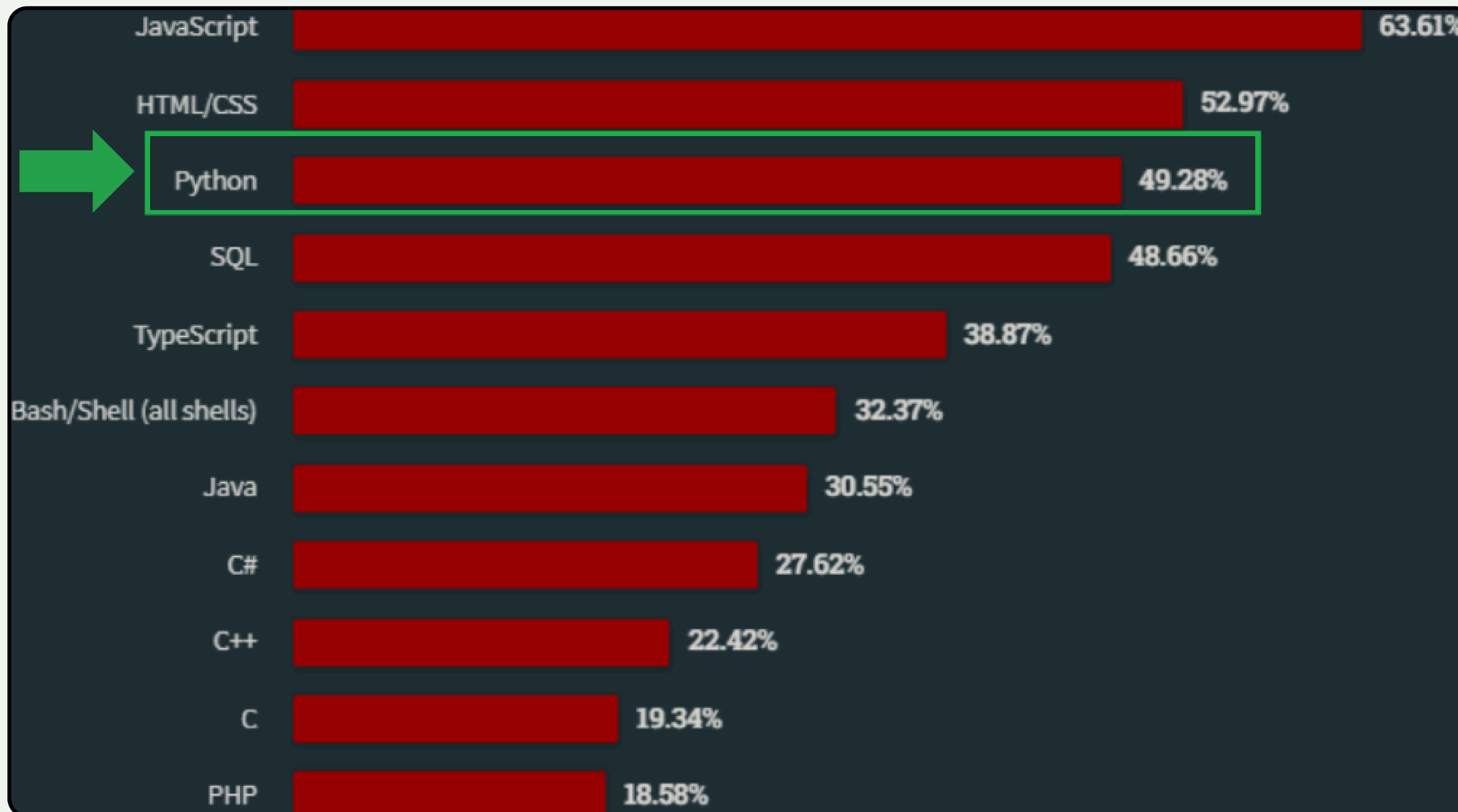
ALGUNOS DATOS ACTUALES...

LevelUP

Tech Academy

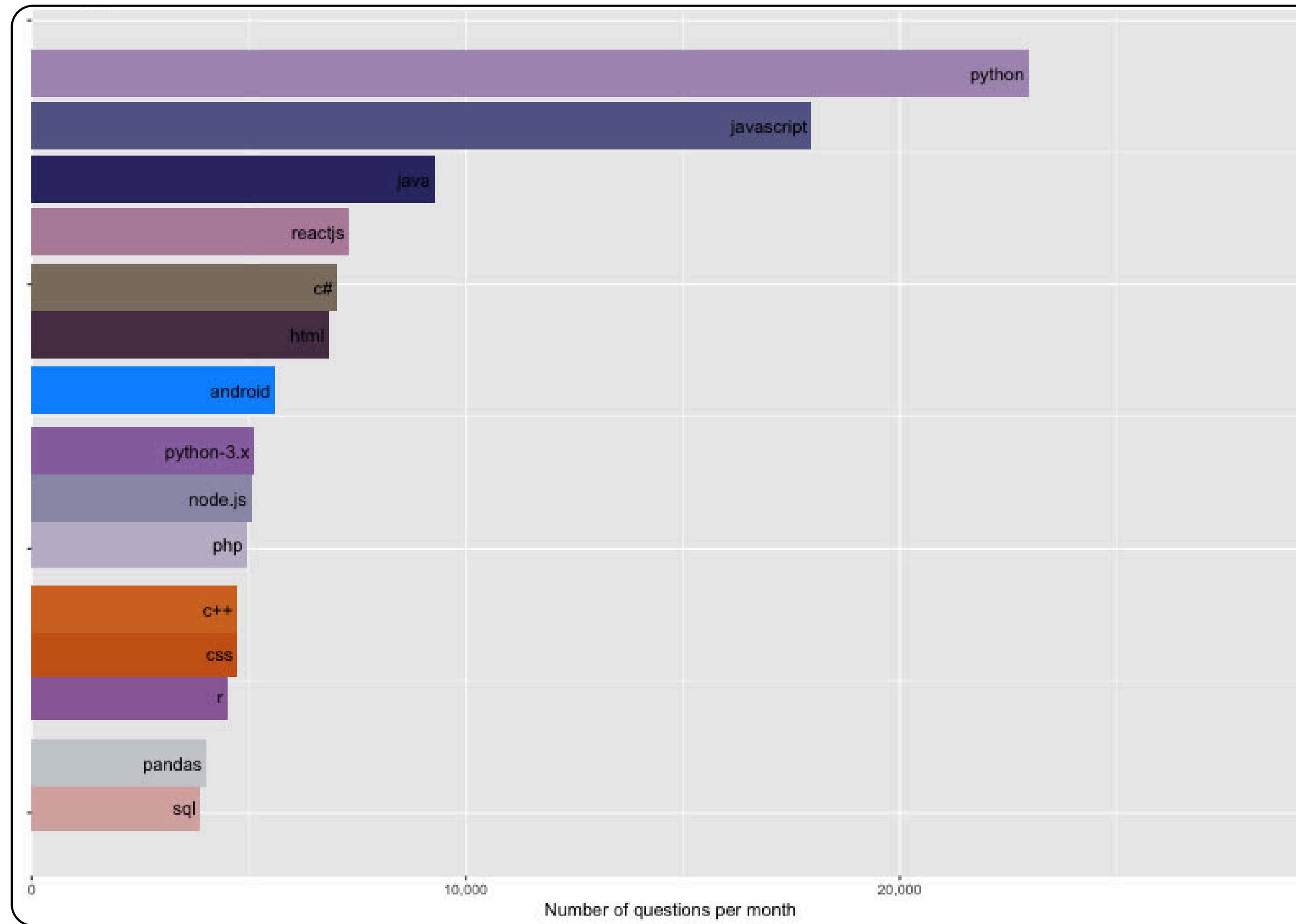
STACK OVERFLOW LENGUAJES MÁS POPULARES 2023

LevelUP



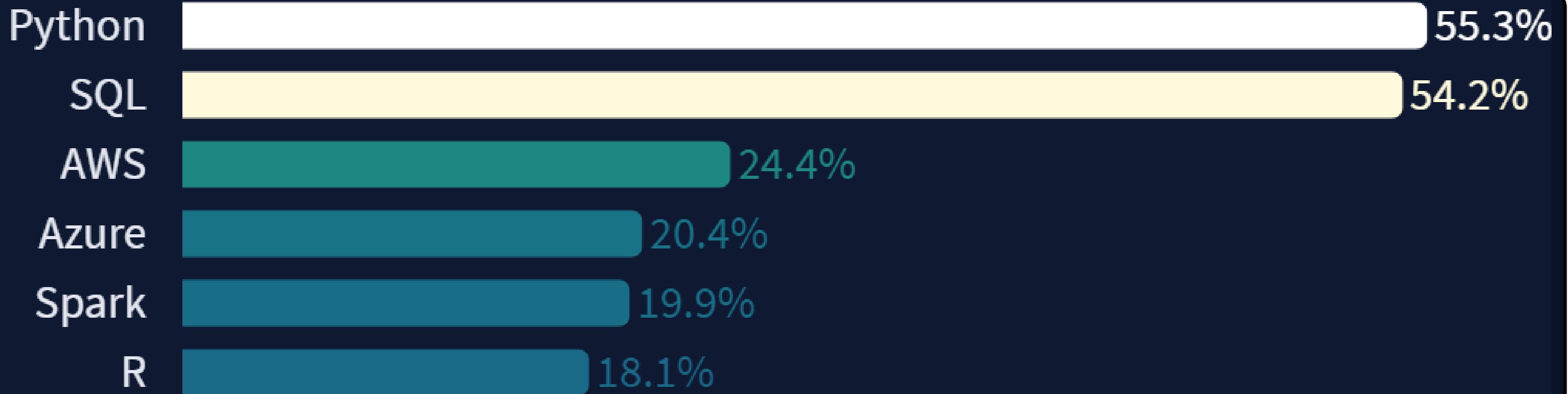
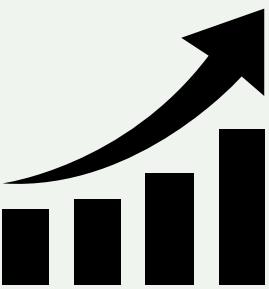
TOP STACK OVERFLOW QUESTIONS

LevelUP

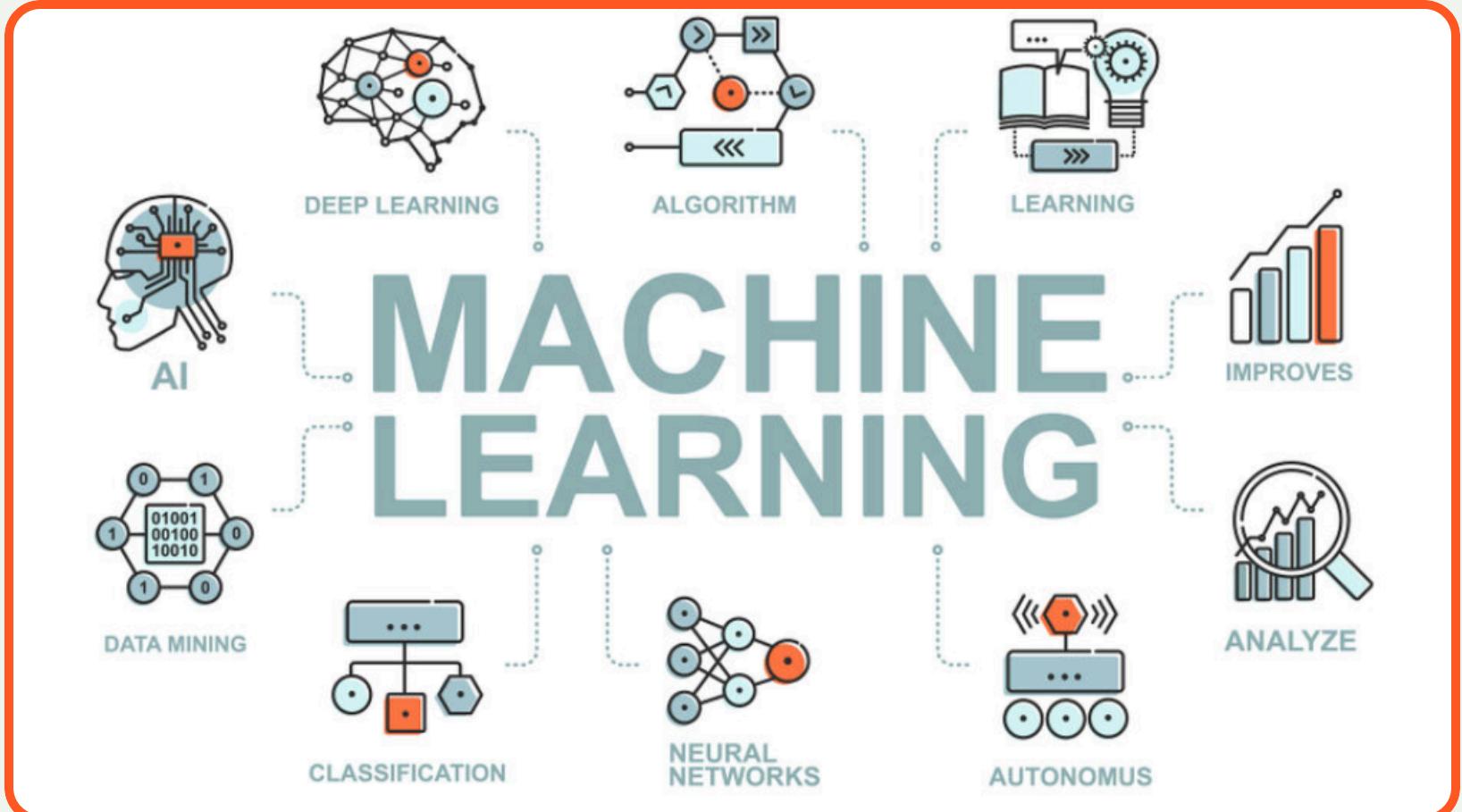


<https://stackoverflow.blog/2023/01/26/comparing-tag-trends-with-our-most-loved-programming-languages/>

TOP DATA ANALISYS SKILLS



¿DÓNDE SE USA PYTHON?

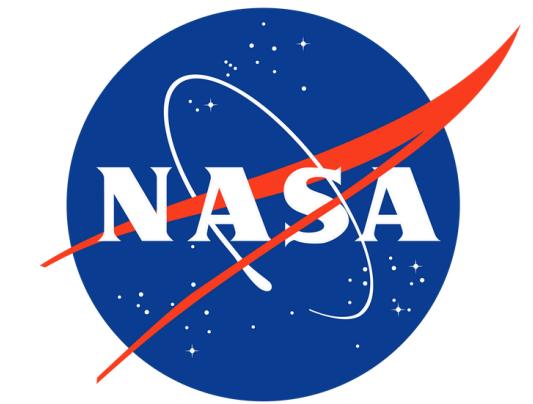


EMPRESAS/APLICACIONES QUE LO USAN

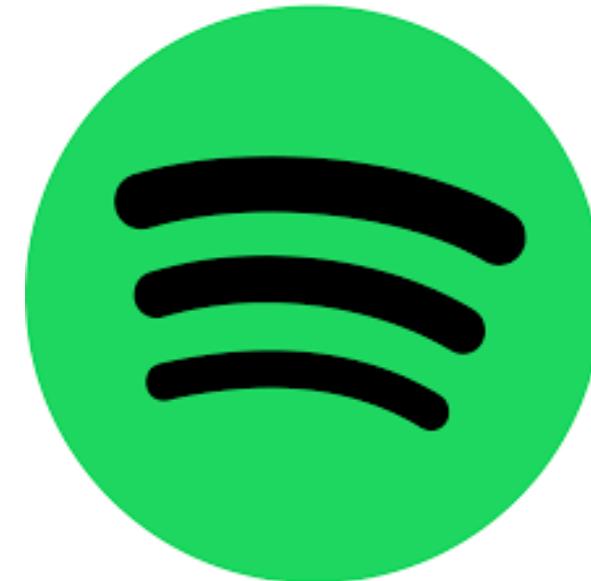
Google



NETFLIX



BATTLEFIELD



amazon

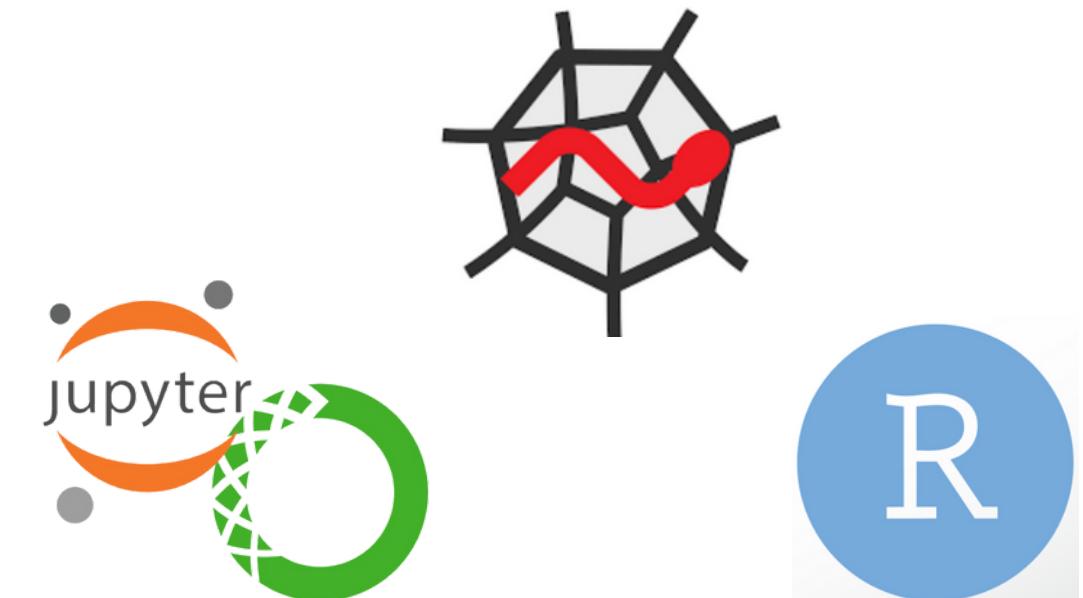
CONCEPTOS INICIALES

ENTORNOS DE DESARROLLO (IDE)

IDE (INTEGRATED DEVELOPMENT ENVIRONMENT)

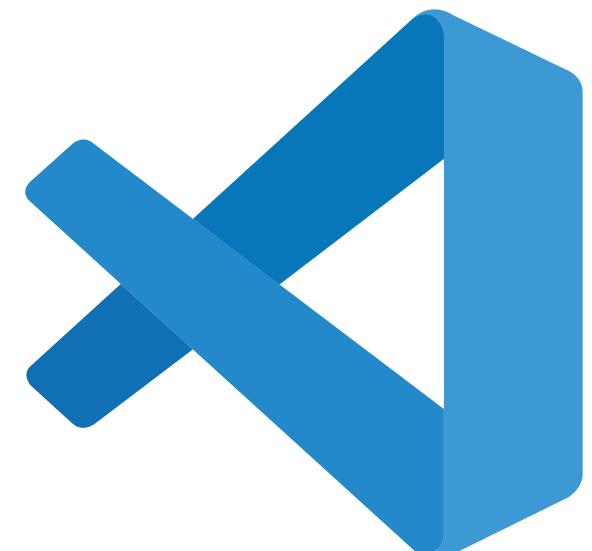
Permiten:

✍ Escribir código



👾 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.

Proporciona acceso a recursos informáticos especialmente para tareas de **Ciencia de Datos y Machine Learning**.

✓ Algunas características:

👉 Ejecución interactiva

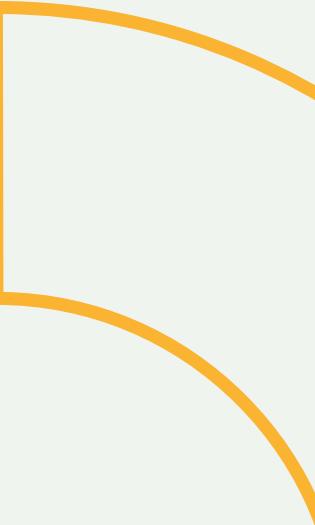
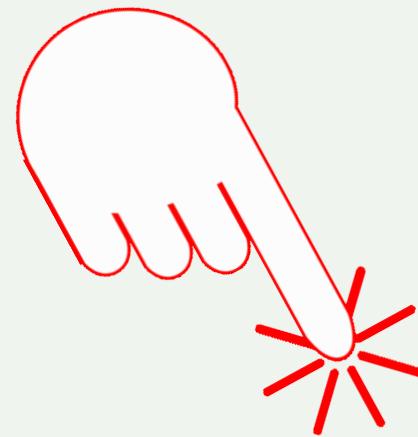
👉 Integración con otras herramientas como google sheets.

Vamos a crear nuestro primer Colab notebook, pero antes...



Mi primer Colab Notebook

LevelUP

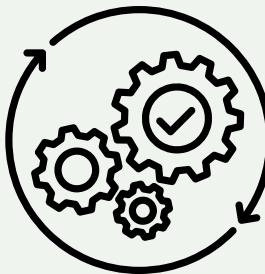




Google Colab
google.com

CONCEPTOS FUNDAMENTALES

¿Qué es Programar ?



Conceptualmente, es indicarle a la computadora qué debe hacer (instrucciones) para realizar una tarea.

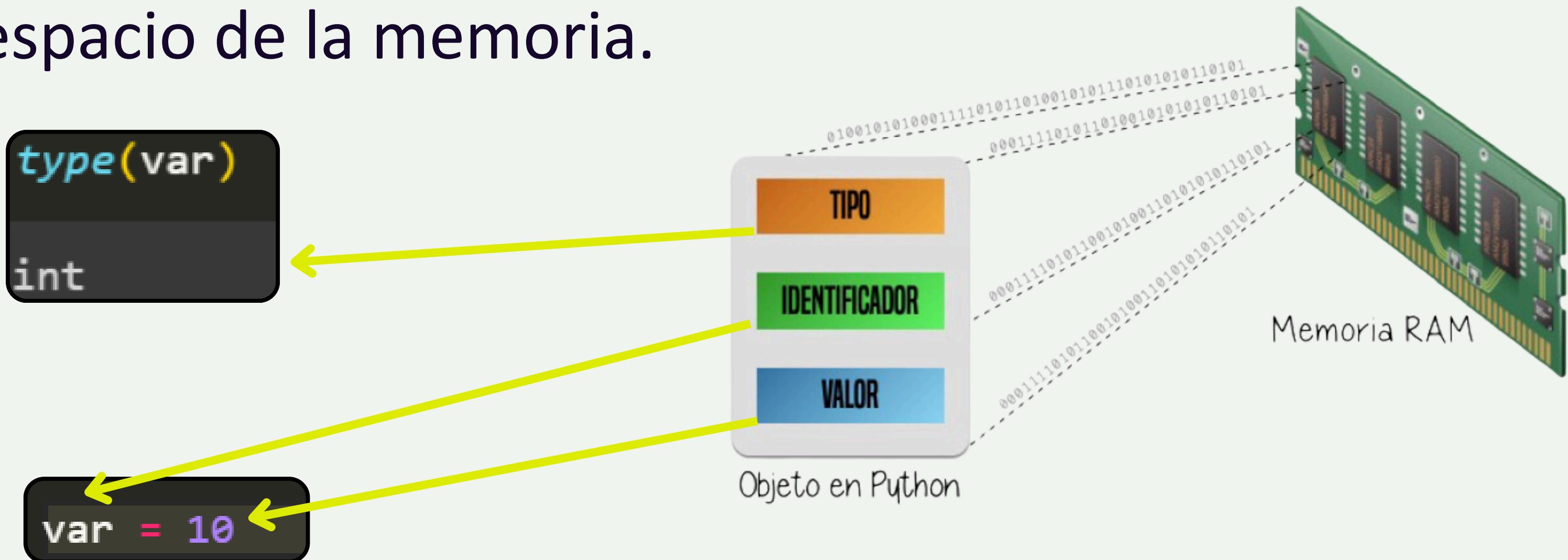
```
31     self._init__(**kwargs)
32     self.file = None
33     self.fingerprints = set()
34     self.logduplicates = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.join(path, 'fingerprints.txt'), 'w')
39         self.file.seek(0)
40         self.fingerprints.update(self.file.read().splitlines())
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getbool('BROWSERZER_DEBUG')
45         return cls(job_dir(settings), debug)
46
47     def request_seen(self, request):
48         fp = self.request_fingerprint(request)
49         if fp in self.fingerprints:
50             return True
51         self.fingerprints.add(fp)
52         if self.file:
53             self.file.write(fp + os.linesep)
```

CODE ROUND 0

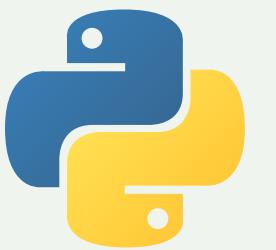
- Introducción

Variables

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



Tipos de datos



LevelUP

- **String:** “Hola LVL UP”
- **Int:** 10
- **Float:** 3.14
- **Bool:** True/False
- **Tuple:** (1,2)

MUTABILIDAD

- **Lista:** [1, ‘Pera’, True, 3.14]
- **Dict:** {‘nombre’:’Guido’}
- **Set:** {1,2,3}

MUCHAS GRACIAS