# Automatiza en 3,2,1…
# con Python

# DISCLAIMER:

# ¡ALERTA DE SESIÓN "SEMI- ABURRIDA"! 

Esta sesión es como ir al gimnasio y solo hacer técnica frente al espejo.
Pero si aguantas esto, mañana estarás escribiendo código que da gusto ver.

Prepárate para:

- Activar entornos
- Versionar con Git sin romper nada
- Escribir código que no te dé pena mostrar

Y recuerda:

**No se construyen automatizaciones profesionales con código improvisado.**

Yo viendo que invite 15 personas a mi curso y llegaron 200


Me da anmsiedad

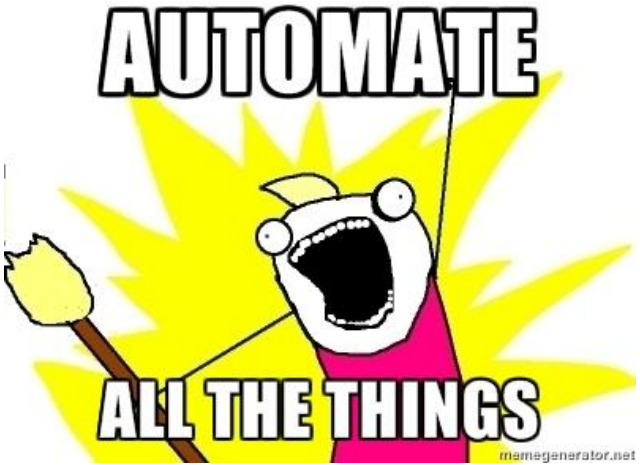
# Sobre mi:

👨‍💻 Ingeniero en Computación
🐍 Desarrollador Backend Python
🏙️ CDMX 🇲🇽
✍️ Zurdo
🌟 Colaboro en:


https://www.meetup.com/python-mexico/events/307557668/
https://es.wikipedia.org/wiki/Usuario_discusi%C3%B3n:Pixelead0

Redes:
**https://github.com/pixelead0**
https://gist.github.com/pixelead0
**https://pixelead0.blogspot.com/**
https://www.linkedin.com/in/pixelead0/
**https://medium.com/@pixelead0**

# Sobre mi:




Mi **superpoder** es **transformar hojas de cálculo** confusas **en archivos JSON** limpios, y hacerlo con código que alguien más pueda entender mañana sin querer prenderme fuego.

He sobrevivido a 11,394 bugs y sigo contando

Firme creyente de que cualquier problema puede resolverse con suficiente **café y Python**

# Sobre mi:


*Aprendí a manejar a mis 41 años,*

*<u>Nunca es tarde para iniciar.</u>*

**Tienes que aprender las reglas del juego.**

**Y luego tienes que jugar mejor que nadie.**

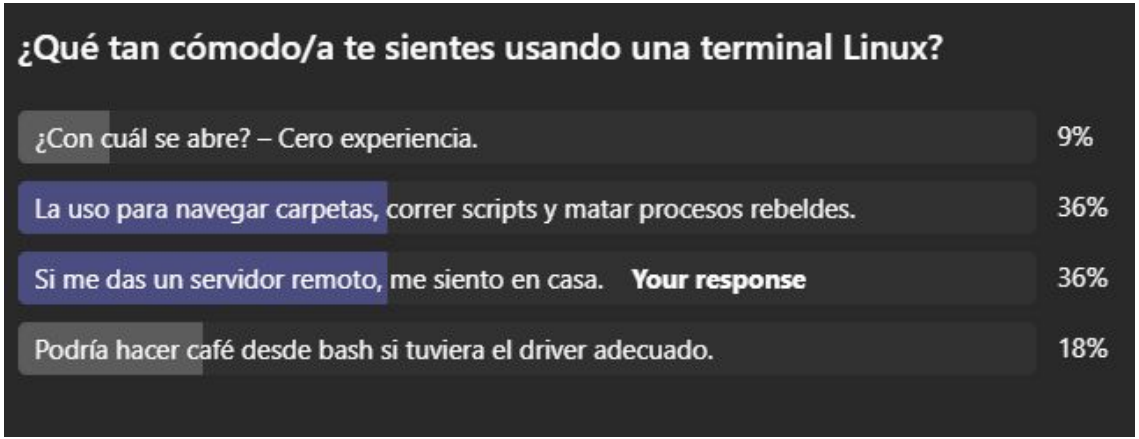
Dianne Feinstein, la primera alcaldesa mujer de San Francisco, durante una entrevista en 1985 para la revista Cosmopolitan

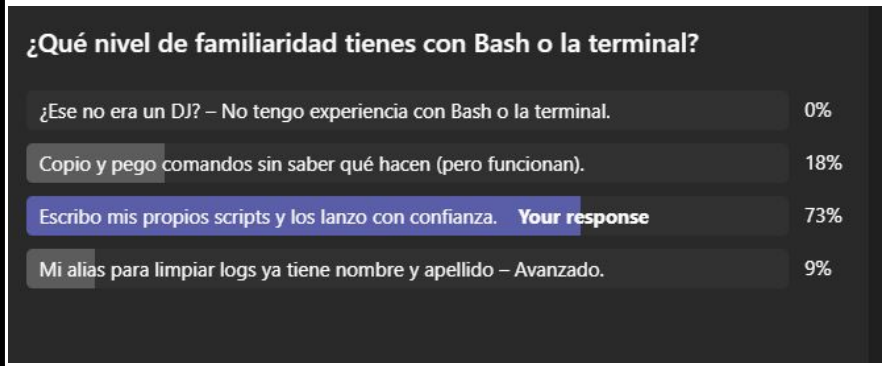
# En nuestro capítulo anterior...

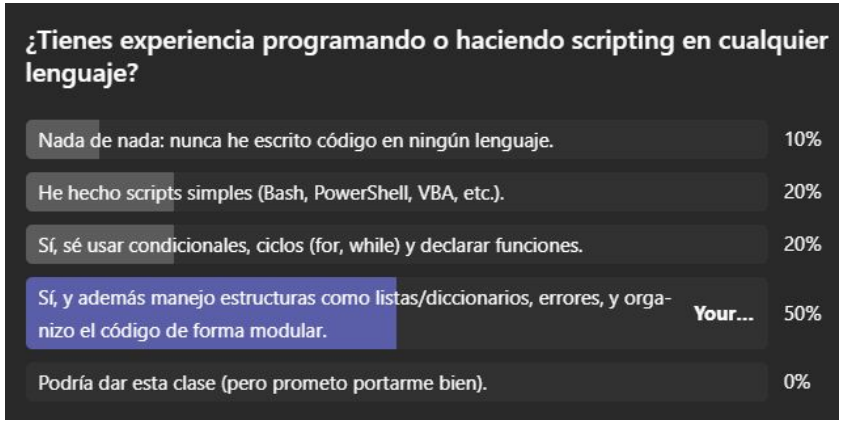
- Historia y filosofía de Python.
- El Zen: escribir código limpio y simple.
- Principios de diseño: SOLID, KISS, DRY

**Por qué automatizar no es solo "que funcione"**

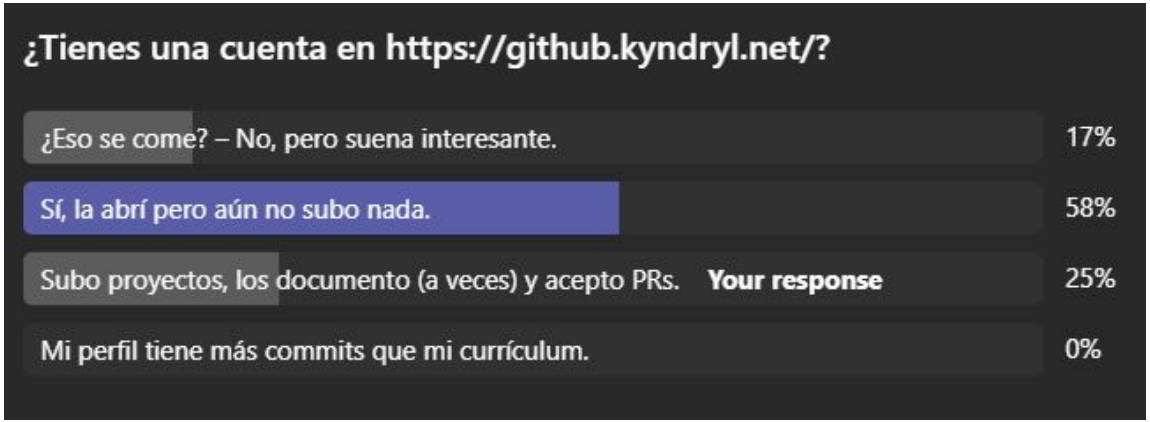
| Name | Status | Score ↓ |
| --- | --- | --- |
| LR Luis Esteban Ronzón (2) | Posted | 10 (100%) |
| IA Isaias Jesus Alcantara (6) | Posted | 10 (100%) |
| J JOSE PEDRO VELAZQUEZ MORALES (5) | Posted | 9 (90%) |
| ER Eliham Miselhem Rios (8) | Posted | 9 (90%) |


¿Qué tan cómodo/a te sientes usando una terminal Linux?
¿Con cuál se abre? – Cero experiencia.
9%
La uso para navegar carpetas, correr scripts y matar procesos rebeldes.
36%
Si me das un servidor remoto, me siento en casa.
Your response
36%
Podría hacer café desde bash si tuviera el driver adecuado.
18%


¿Qué nivel de familiaridad tienes con Bash o la terminal?
¿Ese no era un DJ? – No tengo experiencia con Bash o la terminal.
0%
Copio y pego comandos sin saber qué hacen (pero funcionan).
18%
Escribo mis propios scripts y los lanzo con confianza.
Your response
73%
Mi alias para limpiar logs ya tiene nombre y apellido – Avanzado.
9%


¿Tienes experiencia programando o haciendo scripting en cualquier lenguaje?
Nada de nada: nunca he escrito código en ningún lenguaje.
10%
He hecho scripts simples (Bash, PowerShell, VBA, etc.).
20%
Sí, sé usar condicionales, ciclos (for, while) y declarar funciones.
20%
Sí, y además manejo estructuras como listas/diccionarios, errores, y organizo el código de forma modular.
Your...
50%
Podría dar esta clase (pero prometo portarme bien).
0%


¿Cuál describiría mejor tu relación con Git?
¿Git... como gitano? – Nunca lo he usado.
33%
Nos llevamos bien, aunque a veces se pone celoso con mis ramas – Hago `merge`, `rebase`, PRs y sé salir de conflictos.
Your...
33%
Peleamos seguido, pero lo intento – He hecho `git clone`, `add` y `commit`.
33%
Soy el terapeuta de mis compañeros cuando se rompe Git – Experto/a.
0%


¿Tienes una cuenta en https://github.kyndryl.net/?
¿Eso se come? – No, pero suena interesante.
17%
Sí, la abrí pero aún no subo nada.
58%
Subo proyectos, los documento (a veces) y acepto PRs.
Your response
25%
Mi perfil tiene más commits que mi currículum.
0%

# ¿Qué veremos hoy?

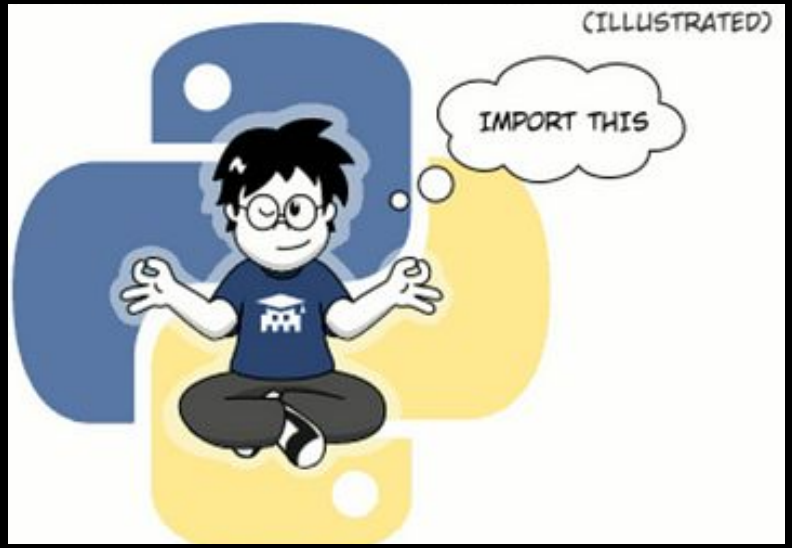
Cómo preparar un entorno profesional

Cómo versionar y estructurar tus scripts

Cómo escribir código reutilizable y legible

Y un ejercicio práctico para cerrar

**Hoy pasamos de la teoría a la acción**



# **Entornos virtuales con** venv

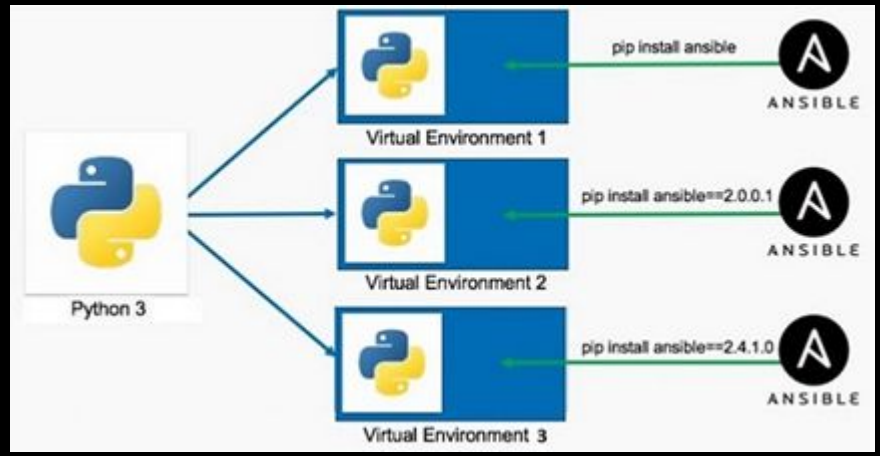
Entender cómo preparar un entorno profesional para programar en Python.

# Entornos Virtuales

Son espacios aislados donde podemos instalar dependencias de Python sin afectar al sistema global ni a otros proyecto

Ventajas:

- Gestionar dependencias específicas para cada proyecto.
- Evitar conflictos entre versiones de paquetes.
- Reproducir fácilmente el entorno de desarrollo en otras máquinas.



# Creación y activación con venv

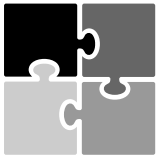
```
# Crear un entorno virtual
python -m venv .venv

# Activar el entorno virtual
# En Linux/Mac:
source .venv/bin/activate

# Verificar activación (el prompt cambia y muestra el entorno)
which python  # Debe mostrar el python del entorno virtual
```

https://docs.python.org/3/tutorial/venv.html

# Creación y activación con venv

```
back-to-the-future@LAPTOP-F1S8CTES:~/python_para_todos$ ls -la
total 8
drwxr-xr-x  2 back-to-the-future back-to-the-future 4096 Apr 19 23:55 .
drwx------ 16 back-to-the-future back-to-the-future 4096 Apr 19 23:53 ..
back-to-the-future@LAPTOP-F1S8CTES:~/python_para_todos$ which python3
/usr/bin/python3
back-to-the-future@LAPTOP-F1S8CTES:~/python_para_todos$ python3 -m venv .venv
back-to-the-future@LAPTOP-F1S8CTES:~/python_para_todos$ ls -la
total 12
drwxr-xr-x  3 back-to-the-future back-to-the-future 4096 Apr 19 23:56 .
drwx------ 16 back-to-the-future back-to-the-future 4096 Apr 19 23:53 ..
drwxr-xr-x  5 back-to-the-future back-to-the-future 4096 Apr 19 23:56 .venv
back-to-the-future@LAPTOP-F1S8CTES:~/python_para_todos$ source .venv/bin/activate
(.venv) back-to-the-future@LAPTOP-F1S8CTES:~/python_para_todos$ which python3
/home/back-to-the-future/python_para_todos/.venv/bin/python3
(.venv) back-to-the-future@LAPTOP-F1S8CTES:~/python_para_todos$
```

# **Gestión de Dependencias** **con** requirements.txt



El archivo **requirements.txt** se utiliza para gestionar las dependencias de un proyecto Python.

Especifica las librerías necesarias para ejecutar el proyecto.

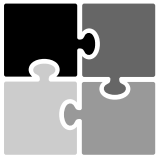
Facilita la instalación de todas las dependencias con un solo comando

```
pip install -r requirements.txt
```

Permite reproducir el entorno de desarrollo fácilmente en diferentes máquinas o entornos.

Es una forma estándar de asegurar que cualquier persona que trabaje en el proyecto tenga las mismas dependencias instaladas.

# **Gestión de Dependencias** **con** requirements.txt



Generar archivo requirements.txt

```
pip freeze > requirements.txt
```

Instalar dependencias

```
pip install -r requirements.txt
```

***Compartir tu código sin requirements.txt es como entregar una receta sin ingredientes.***

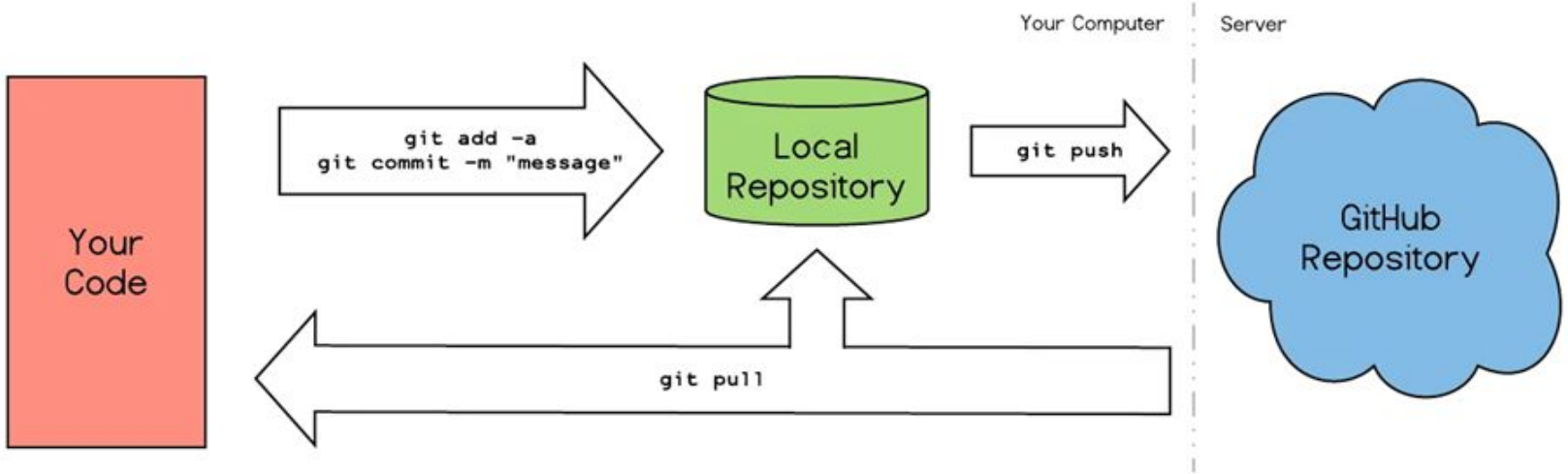
# Entornos Virtuales

**“Tu entorno es tu laboratorio.**
**No lo contamines.”**

# CONTROL DE VERSIONES

El control de versiones es una herramienta esencial para todo desarrollador.
Permite registrar, comparar, revertir y colaborar en los cambios de código a lo largo del tiempo.

# Control de Versiones




*“Automatizar sin control de versiones… es vivir al borde del abismo.”*

# Comandos esenciales:

git pull

git status

git add .

git commit -m "feat: agregar validador de IPs"

git branch

git push

# Commits Convencionales

Son una especificación para dar estructura a los mensajes de commit en Git. Esta convención proporciona un conjunto fácil de reglas para crear un historial de commits explícito, lo que facilita la creación de herramientas automatizadas y la generación de changelogs.

**VENTAJAS:**

Generación automática de changelogs

Determinación automática de incrementos de versión semántica

Comunicar la naturaleza de los cambios a compañeros de equipo y otros interesados

Facilitar la contribución a proyectos al estandarizar la estructura de commits

Hacer más útil el historial de commits y facilitar la navegación

# Control de Versiones


AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.



https://imgs.xkcd.com/comics/git_commit.png

# Commits Convencionales

Tips principales:

| | |
|---|---|
| **Feat** | Nueva característica |
| **Docs** | Cambios en documentación |
| **Style** | Cambios que no afectan al significado del código (espaciado, formato, etc.) |
| **Refactor** | Cambio de código que no corrige error ni añade funcionalidad |
| **Perf** | Mejora de rendimiento |
| **Test** | Añadir o corregir pruebas |
| **Build** | Cambios en sistema de build o dependencias externas |
| **Chore** | Otros cambios que no modifican src o test |

# Control de Versiones


"Esto es GIT. Rastrea el trabajo colaborativo en proyectos a través de un hermoso modelo de árbol basado en teoría de grafos distribuidos."

"Genial. ¿Cómo lo usamos?"

"Ni idea. Solo memoriza estos comandos de terminal y escríbelos para sincronizar. Si obtienes errores, guarda tu trabajo en otro lugar, borra el proyecto, clónalo de nuevo y descarga una copia nueva."



https://img.devrant.com/devrant/rant/r_2329045_c5v7p.jpg

# Recordatorio: Tomar agua.

# ¿Por qué importa la estructura?

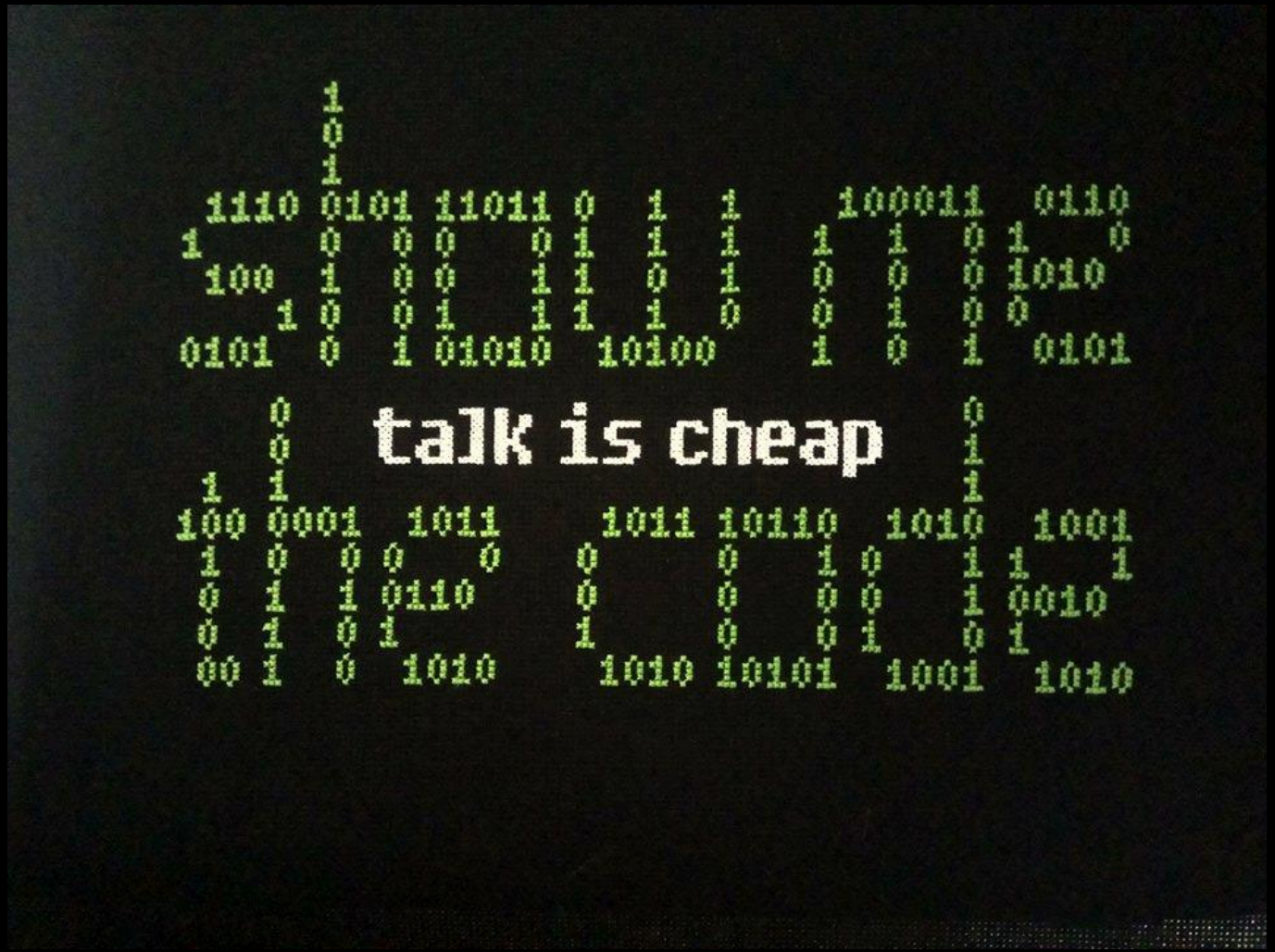
- Facilita la lectura y el mantenimiento
- Permite separar responsabilidades
- Es la base para crecer con orden

# Ejemplo

```
proyecto/
├── .venv/
├── main.py
├── helpers.py
├── requirements.txt
└── .gitignore
```

*Bonus:*

*incluye README.md y GUIDE.md si tu proyecto crece*


talk is cheap

# EJERCICIO PRÁCTICO

**GENERADOR DE CONTRASEÑAS**

Objetivo:

Crear una función que genere contraseñas aleatorias y seguras con opciones personalizadas.

# EXTENSIÓN DEL EJERCICIO

**FILTROS Y OPCIONES**

Objetivo:

Agregar flexibilidad al generador de contraseñas.

# MEJORA:

**LONGITUD PERSONALIZADA CON input()**

Objetivo:

Permitir al usuario definir la longitud de la contraseña desde la terminal.

# INTERACTIVO:

**OPCIONES PERSONALIZADAS CON input()**

Objetivo:

Pedir al usuario dos entradas:

- Longitud de la contraseña
- Si desea incluir caracteres especiales

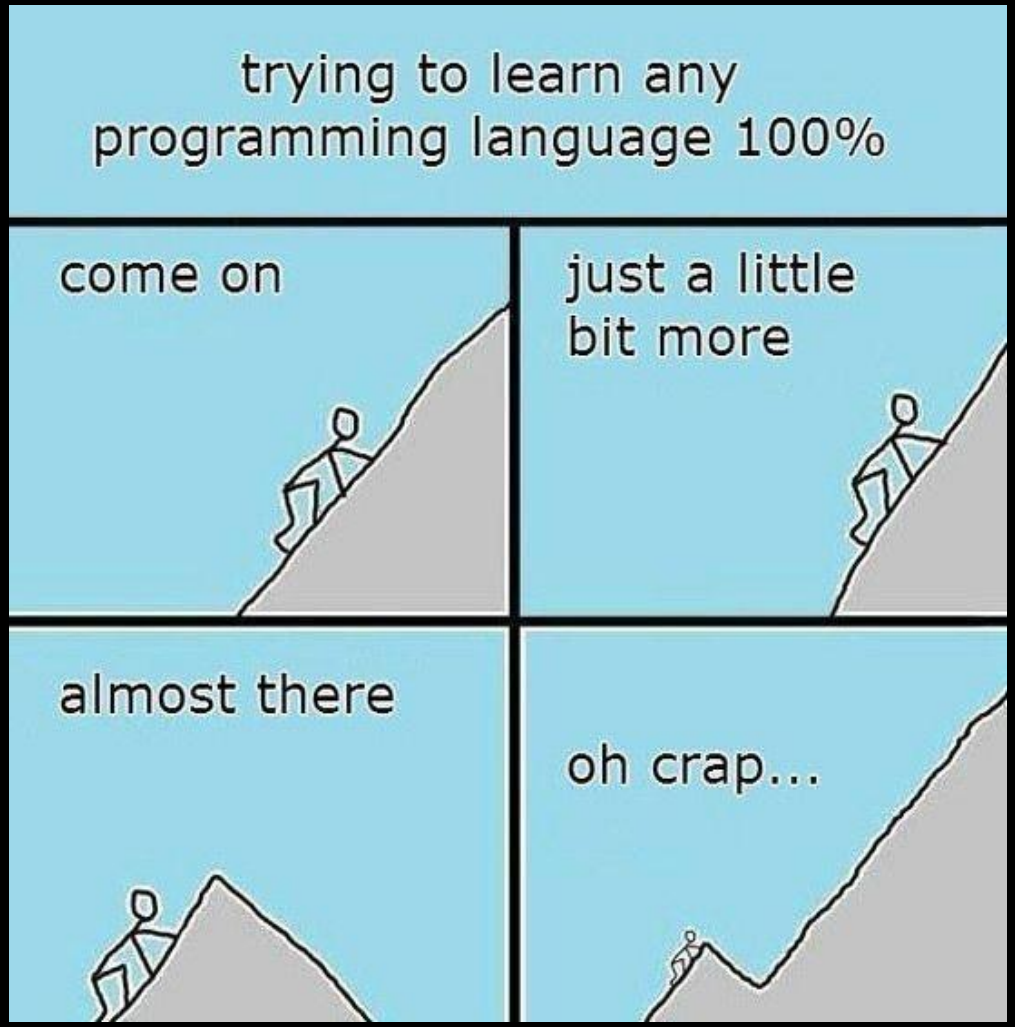
# ¿Qué aprendimos hoy?

✔️Crear entornos virtuales con venv
✔️ Versionar el código con Git
✔️ Estructurar un proyecto Python
✔️ Escribir funciones reutilizables
✔️ ¡Y construir un generador de contraseñas pro!

# Reto opcional

Guarda las contraseñas generadas en un archivo .txt

O crea una opción para generar varias contraseñas de golpe


trying to learn any programming language 100%
come on
just a little bit more
almost there
oh crap...

# ¿Qué sigue?

Validaciones, manejo de errores y pruebas automáticas

Para que tu script funcione siempre… incluso cuando todo sale mal.


GRACIAS POR
SU ATENCIÓN
BUENO A LOS QUE
PRESTARON ATENCIÓN!
memegenerator.es