

# Curso Python

## Feb-Mar

## 2024



# Sobre este curso:

- ✓ Entenderemos como es un entorno de trabajo en Python
- ✓ Ampliaremos conocimientos sobre la librería estándar
- ✓ Obtendremos herramientas útiles para nuestros proyectos
- ✓ Seremos capaces de interpretar más tipos de estructuras programáticas

```
@lru_cache
def fib(n):
    if n <= 1:
        return n
    else:
        return fib(n-2) + fib(n-1)
```



# Contenidos:

## Entorno de trabajo

- PEP8
- venv
- Jupyter
- Git\*

## Librería estándar

- Mutabilidad
- Funciones
- NamedTuples
- Otros\*

## Librerías de terceros

- Pandas
- Matplotlib\*



# PEP8: ¿Qué es?



- Guía de estilo para la implementación de Python en C (C-Python)
- “Readability counts!”
- La consistencia es importante
- “El código se lee más de lo que se escribe” - Guido van Rossum

# PEP8: Ejemplos

# Wrong:

```
spam( ham[ 1 ], { eggs: 2 } )
```

# Wrong:

# operators sit far away from their operands

```
income = (gross_wages +  
          taxable_interest +  
          (dividends - qualified_dividends) -  
          ira_deduction -  
          student_loan_interest)
```

# Correct:

```
spam(ham[1], {eggs: 2})
```

# Correct:

# easy to match operators with operands

```
income = (gross_wages  
          + taxable_interest  
          + (dividends - qualified_dividends)  
          - ira_deduction  
          - student_loan_interest)
```

# PEP8: ¿Indiscutible?

“A Foolish Consistency is the Hobgoblin of Little Minds”

- Hay que ser inteligente
- Mira ejemplos y ¡pregunta!
- Juzga lo que crees que es más conveniente
- No rompas compatibilidad de versiones anteriores



# PEP8: ¿Indiscutible?

Pero, si no tienes un buen motivo para romper las reglas:

- PEP8



# Entorno de trabajo: Herramientas





# Entorno de trabajo: Vscode

Extensiones:

- vscode-icons
- Jupyter (Extensión Pack (4))
- Python



# Entorno de trabajo: Git y Github



## Git

- Controlador de versiones
- Disponible en gran cantidad de distribuciones Linux

## Github

- Plataforma de alojamiento de repositorios
- Se sirve de Git para el control de versiones

# Entorno de trabajo: Git y Github

Comandos básicos:

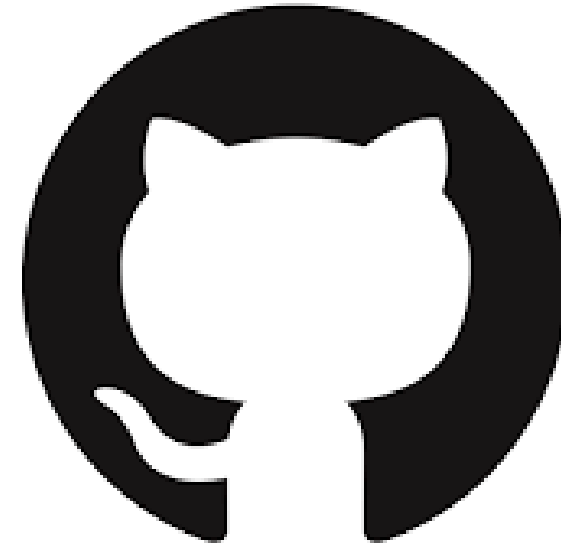
- `git status`
- `git add <file_name>`
- `git commit -m "<message>"`
- `git push`



# Entorno de trabajo: Git y Github

Archivos estructura de un proyecto:

- README.md
- .gitignore
- requirements.txt (pip)
- Otros



# Entorno de trabajo: Jupyter

Tipos de celdas:

- Markdown
- Code



# Entorno de trabajo: Jupyter Markdown

Ejemplos:

```
1 # Title 1
2 ## Title 2
3
4 - Item 1
5 - Item 2
6   - Subitem 1
7   - Subitem 2
8
9 ```python
10 print('Hello world')
11 ```
12 No line
13 jump
14
15 Line\
16 jump
17
```

Title 1

Title 2

- Item 1
- Item 2
  - Subitem 1
  - Subitem 2

```
print('Hello world')
```

No line jump

Line  
jump

```
1 $e^i - 1 = 0$
```

$$e^i - 1 = 0$$

# Entorno de trabajo: venv

Virtual enviroment:

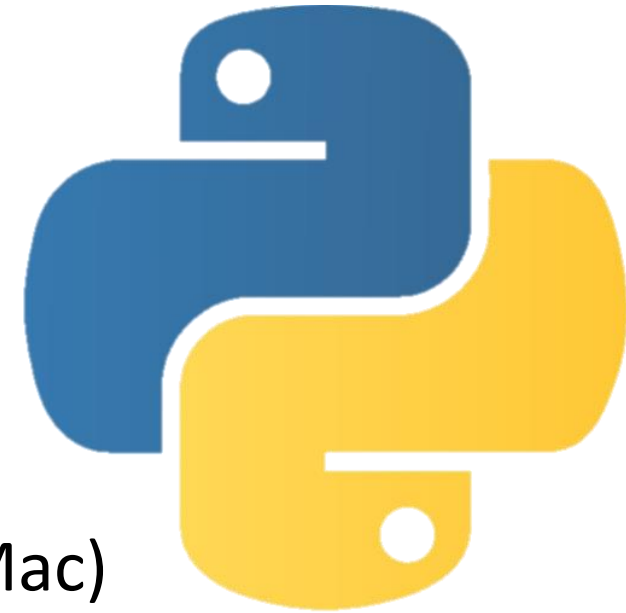
- Modular
- Copia de nuestro Python global
- Ahí instalaremos librerías



# Entorno de trabajo: venv

Comandos:

- `python -m venv .venv`
- `.venv\Scripts\activate` (Windows)
- `source .venv/Scripts/actíivate` (Linux y Mac)





# Entorno de trabajo: venv y pip

Una vez tengamos activo nuestro venv:

- `pip install <module/package>`
- `pip install -r requirements`



# Entorno de trabajo: Vim/Neovim\*

Mención especial:

- Editor de texto potente
- Lo que se usa a nivel profesional



# Entorno de trabajo: Github Codespaces

Trabajo en la nube:

- Entorno de trabajo en el navegador
- Vscode por defecto
- Más potente Vscode local

