



## Exercici 1

### 1- Amb les teves pròpies paraules, podries explicar que es,?

- a. **Jupyter:** És una interfície que permet crear, compartir i executar codi en un format anomenat "notebook". Es pot combinar codi, text explicatiu (markdown), gràfics i sortides de programes en un sol document.
- b. **Anaconda:** És una plataforma que inclou tot el necessari per treballar amb Python o R, especialment en ciència de dades. Simplifica la instal·lació de biblioteques i gestiona entorns per evitar problemes amb dependències.
- c. **Paquet logging:** És una funcionalitat integrada de Python que registra esdeveniments del programa, com errors o estats del sistema. Ajuda a monitoritzar i depurar aplicacions de manera eficient.

## Exercici 2

### 2- Per que creus que es necessari implementar un log en un programa? Raona la resposta i fica exemples del perquè ho creus (mínim 3 exemples).

Implementar un log en un programa és essencial perquè permet fer un seguiment del que està passant durant l'execució del codi. Això és útil per identificar problemes, entendre el comportament del programa i millorar el seu rendiment. A continuació, et dono tres exemples per explicar-ho:

#### 1. Detectar i solucionar errors inesperats

Quan un programa falla, el log pot mostrar on i per què ha passat l'error. Per exemple:

Si un servidor web dona un error 500, el log pot indicar que és perquè falta una base de dades o hi ha un error en la consulta SQL.

Exemple de log:

ERROR: No s'ha pogut connectar a la base de dades: "Connection timed out".

#### 2. Monitoritzar el funcionament del sistema

Els logs permeten veure si tot funciona correctament o si hi ha problemes que encara no han afectat l'usuari. Exemple: Un programa que gestiona comandes d'una botiga en línia pot registrar cada comanda amb un missatge del tipus:

INFO: Comanda 123 processada correctament.

WARNING: Comanda 124 ha trigat més de 5 segons en processar-se.

#### 3. Auditoria i seguiment de decisions

Els logs també serveixen per entendre decisions preses pel sistema, sobretot en aplicacions crítiques o financeres.

Per exemple, un sistema bancari pot enregistrar transaccions amb detalls com:

INFO: Transacció de 200€ aprovada per l'usuari 456.

CRITICAL: Transacció fallida per saldo insuficient.

## Exercici 3

**Amb les teves paraules, explica que significa cada línia dels següents exemples de logging.**

### Logging 1

```
logging.basicConfig(format='%(asctime)s - %(message)s', level=logging.INFO)
logging.info('Admin ha accedido')
```

1. `logging.basicConfig(format='%(asctime)s - %(message)s', level=logging.INFO)`
  - Configura com es mostraran els missatges de log.
  - `format='%(asctime)s - %(message)s'`: Estableix el format dels logs, on:
    - `%(asctime)s`: Afegeix la data i hora del moment en què es registra el missatge.
    - `%(message)s`: Mostra el missatge que l'usuari defineix.
  - `level=logging.INFO`: Especifica que només es registraran missatges de nivell INFO o superior (com WARNING, ERROR, etc.).
2. `logging.info('Admin ha accedido')`
  - Escriu un missatge de nivell INFO amb el text "Admin ha accedido". Si el programa arriba aquí, el log mostrarà una línia semblant a:  
2024-11-21 12:34:56,789 - Admin ha accedido  
Això indica quan l'administrador ha accedit al sistema.

### Logging 2

```
f_handler = logging.FileHandler('file.log')
```

1. `f_handler = logging.FileHandler('file.log')`
  - Crea un gestor de fitxers (FileHandler) que guardarà els logs en un fitxer anomenat file.log.
  - Això permet que els registres no es mostrin només a la consola, sinó que quedin emmagatzemats per a futures referències, com depuració o auditories.

### Logging 3

```
import logging
```

```
logging.basicConfig(format='%(process)d-%(levelname)s-%(message)s')
logging.warning('This is a Warning')
```

1. `import logging`
  - Importa la biblioteca de logging per poder fer ús de les seves funcionalitats.
2. `logging.basicConfig(format='%(process)d-%(levelname)s-%(message)s')`
  - Configura el format dels missatges de log.
  - `%(process)d`: Inclou l'identificador del procés que ha generat el missatge (important en aplicacions amb múltiples processos).
  - `%(levelname)s`: Mostra el nivell del log (per exemple, WARNING, ERROR, etc.).
  - `%(message)s`: Inclou el missatge de log definit per l'usuari.
3. `logging.warning('This is a Warning')`
  - Escriu un missatge de nivell WARNING amb el text "This is a Warning". El resultat seria una línia semblant a:  
12345-WARNING-This is a Warning  
Aquí, 12345 representa l'identificador del procés. El log indica que hi ha hagut un avís que caldria revisar.

## Exercici 4

A partir del següent codi creat amb JUPYTER explica resumidament el que fa cada BLOC. No cal explicar línia per línia, sinó que pot ser de forma general, però si amb certa coherència.

### Taula d'exemple

```
import pandas as pd
data = pd.DataFrame({'Profesor': ['Emili', 'Isabel', 'Jose', 'Emili', 'Isabel'],
                    'Alumno': ['Xavier', 'Juan', 'Carlos', 'Maradona', 'Paquito'],
                    'M01': ['8', '6', '5', '5', '5'],
                    'M02': ['6', '4', '8', '4', '9'],
                    'M03': ['7', '5', '7', '3', '8'],
                    'M04': ['9', '7', '4', '8', '9'],
                    'M05': ['4', '5', '7', '8', '4']})
```

	Profesor	Alumno	M01	M02	M03	M04	M05
0	Emili	Xavier	8	6	7	9	4
1	Isabel	Juan	6	4	5	7	5
2	Jose	Carlos	5	8	7	4	7
3	Emili	Maradona	5	4	3	8	8
4	Isabel	Paquito	5	9	8	9	4

### Codi 1

```
notas[["M01", "M02", "M03", "M04", "M05"]].mean()
```

```
M01    5.8
M02    6.2
M03    6.0
M04    7.4
M05    5.6
dtype: float64
```

Aquest bloc de codi fa el següent:

1. **Selecció de columnes numèriques:**
  - Es seleccionen les columnes del DataFrame que contenen les notes dels mòduls: **M01**, **M02**, **M03**, **M04** i **M05**.
2. **Càlcul de la mitjana:**
  - La funció `.mean()` es calcula sobre aquestes columnes. Aquesta funció calcula la **mitjana aritmètica** dels valors de cada columna. Com que les notes en el teu exemple són valors en format text (strings), aquest codi donaria un error tret que abans s'hagin convertit a valors numèrics.

### Codi 2

```
[3]: notas.groupby('Alumno', as_index=False)[["M01", "M02", "M03", "M04", "M05"]].mean()
```

```
[3]:
```

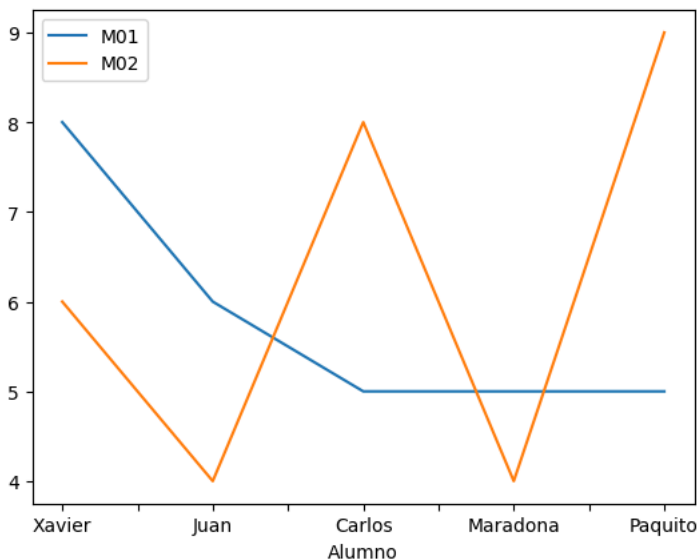
	Alumno	M01	M02	M03	M04	M05
0	Carlos	5.0	8.0	7.0	4.0	7.0
1	Juan	6.0	4.0	5.0	7.0	5.0
2	Maradona	5.0	4.0	3.0	8.0	8.0
3	Paquito	5.0	9.0	8.0	9.0	4.0
4	Xavier	8.0	6.0	7.0	9.0	4.0

1. **Agrupació per alumne:**

- S'utilitza la funció `.groupby('Alumno', as_index=False)` per agrupar les dades del DataFrame segons el nom dels alumnes, assegurant que la columna 'Alumno' es mantingui com a columna en lloc d'index.
- 2. **Selecció de columnes de notes:**
  - Es treballa només amb les columnes corresponents als mòduls: **M01**, **M02**, **M03**, **M04** i **M05**.
- 3. **Càlcul de la mitjana:**
  - Sobre cada grup (cada alumne), es calcula la mitjana de les seves notes en els diferents mòduls mitjançant `.mean()`.

### Codi 3

```
import matplotlib.pyplot as plt
notas.plot(x="Alumno", y=["M01", "M02"])
plt.rcParams["figure.figsize"] = (10,5)
plt.show()
```



Aquest bloc de codi genera un **gràfic** que visualitza les notes dels alumnes en dos mòduls (**M01** i **M02**), amb una configuració per ajustar la mida del gràfic. A continuació, s'explica què fa cada part de manera general:

1. **Creació del gràfic:**
  - `notas.plot(x="Alumno", y=["M01", "M02"])` genera un gràfic a partir del DataFrame `notas`, utilitzant:
    - La columna "**Alumno**" com a eix X (categories representades pels noms dels alumnes).
    - Les columnes "**M01**" i "**M02**" com a valors a l'eix Y (les notes d'aquests mòduls).
2. Per defecte, `plot()` crea un gràfic de línies, però aquest comportament pot variar segons el context (per exemple, si s'especifica un altre tipus de gràfic).
3. **Configuració de la mida del gràfic:**
  - `plt.rcParams["figure.figsize"] = (10, 5)` ajusta la mida del gràfic a 10 d'ample i 5 d'alt (en unitats predeterminades).
4. **Mostra del gràfic:**
  - `plt.show()` mostra el gràfic generat, útil especialment en entorns interactius com Jupyter Notebook.

### Codi 4

```
[4]: notas[["M01", "M02", "M03", "M04", "M05"]].mean(axis = 1)

[4]: 0    6.8
      1    5.4
      2    6.2
      3    5.6
      4    7.0
      dtype: float64
```

Aquest codi calcula la mitjana de les notes de cada alumne a través de tots els mòduls (**M01** a **M05**) i té els següents passos generals:

**1. Selecció de columnes de notes:**

- Es treballa amb les columnes de notes dels mòduls: **M01, M02, M03, M04 i M05**.

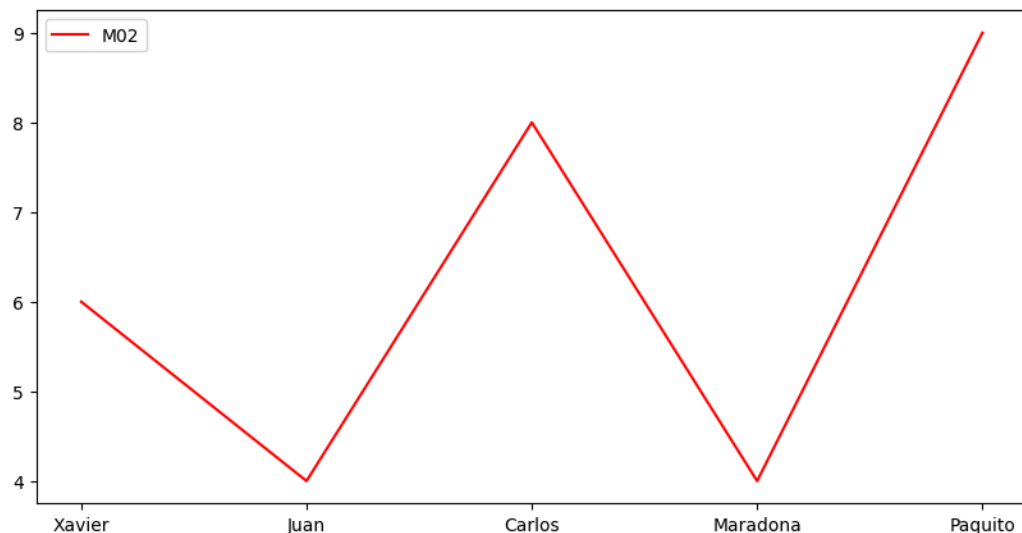
**2. Càlcul de la mitjana per fila:**

- La funció `.mean(axis=1)` calcula la mitjana a nivell de fila (per cada alumne). Això es fa especificant `axis=1`, que indica que l'operació s'ha de realitzar horitzontalment, és a dir, a través de les columnes seleccionades.

## Codi 5

```
x_values=notas["Alumno"]
y_values=notas["M02"]
plt.plot(x_values, y_values, color="red")
plt.legend(["M02"])
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Aquest bloc de codi crea un **gràfic de línies** per mostrar les notes dels alumnes en el mòdul **M02**, amb una llegenda per identificar la línia. Aquí tens una explicació general:

**1. Definició de les dades per al gràfic:**

- `x_values = notas["Alumno"]`: S'assignen els noms dels alumnes (de la columna "**Alumno**") a la variable `x_values`. Aquestes dades es faran servir per a l'eix X del gràfic.
- `y_values = notas["M02"]`: S'assignen les notes dels alumnes en el mòdul **M02** (de la columna "**M02**") a la variable `y_values`. Aquestes dades es faran servir per a l'eix Y del gràfic.

**2. Creació del gràfic:**

- `plt.plot(x_values, y_values, color="red")`: Crea un gràfic de línies on les dades dels alumnes es col·loquen en l'eix X i les seves notes de **M02** en l'eix Y. La línia es dibuixa de color vermell (`color="red"`).

**3. Afegir llegenda:**

- `plt.legend(["M02"])`: Afegeix una llegenda que indica que la línia mostra les notes del mòdul **M02**.

**4. Mostrar el gràfic:**

- `plt.show()`: Mostra el gràfic generat en pantalla.

## Exercici 5

**Creus que aquesta pàgina web, d'una marca molt coneguda, ha estat dissenyada utilitzant els principis/regles del "disseny d'interfícies d'usuaris"?**

**1- Analitza la pàgina de l'enllaç de sota i raona per quins motius has elegit "si o no" a la pregunta anterior. (La resposta es oberta, es a dir que a part de les regles, també pots per exemple explicar si els colors de la web son els mes adients etc...)**