

CURS PROGRAMACIÓ AMB PYTHON



ALUMNE: ALFREDO DÍAZ

CONDICIONALES

```
1 calificacion = int(input("Introduce el tipo de calificacion(1=detallada, 2=breve):"))
2 nota = float(input("Introduce una nota (1-10):"))
3 if nota not in range(1, 10):
4     print("Numero de nota incorrecto")
5 else:
6     if calificacion == 1:
7         if nota >=9:
8             print("Sobresaliente")
9         elif nota >=7:
10            print("Notable")
11        elif nota >=6:
12            print("Bien")
13        elif nota >=5:
14            print("Suficiente")
15        else:
16            print("Insuficiente")
17
18    elif calificacion == 2:
19        if nota >=9:
20            print("Excelente")
21        elif nota >= 4.9:
22            print("Apto")
23        else:
24            print("No Apto")
25
26    else:
27        print("Numero de calificacion incorrecto")
```

Introduce el tipo de calificacion(1=detallada, 2=breve):1
Introduce una nota (1-10):9
Sobresaliente

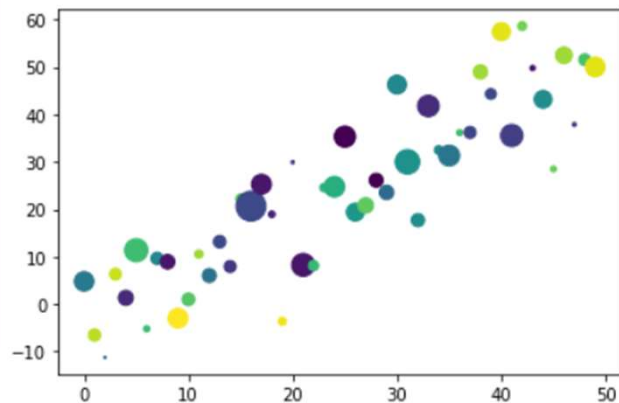
COLECCIONES/DICCIONARIOS

```
1 import random
2
3 aleatoris = ["dilluns", "dimarts", "dimecres", "dijous", "divendres",
4             "dissabte", "diumenge"]
5 random.shuffle(aleatoris)
6
7 print (aleatoris)
8
9 dic_fr = {"dilluns": "lundi", "dimarts": "mardi", "dimecres": "mercredi",
10          "dijous": "jeudi", "divendres": "vendredi",
11          "dissabte": "samedi", "diumenge": "dimanche"}
12 for d in aleatoris :
13     print (f" el dia {d:10} traduït és {dic_fr[d]}")
```

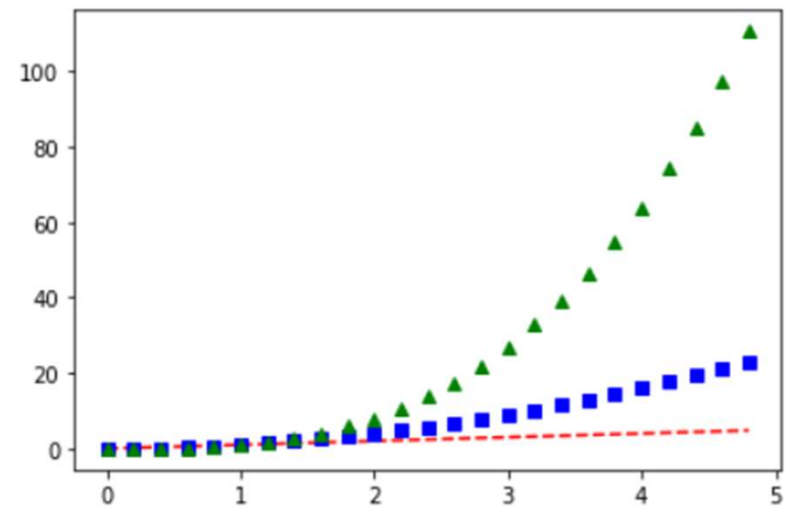
['dijous', 'diumenge', 'dimecres', 'divendres', 'dilluns', 'dimarts', 'dissabte']
el dia dijous traduït és jeudi
el dia diumenge traduït és dimanche
el dia dimecres traduït és mercredi
el dia divendres traduït és vendredi
el dia dilluns traduït és lundi
el dia dimarts traduït és mardi
el dia dissabte traduït és samedi

MATPLOTLIB

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 np.random.seed(19680801)
5
6 data = {'a': np.arange(50),
7         'c': np.random.randint(0,50,50),
8         'd': np.random.randn(50)}
9
10 data['b'] = data['a'] + 10 * np.random.randn(50)
11 data['d'] = np.abs(data['d']) * 100
12
13 plt.scatter('a','b',c='c', s='d', data=data)
14 plt.xlabel = 'entry a'
15 plt.ylabel = 'entry b'
16 plt.show()
```



```
1 import matplotlib.pyplot as plt
2
3 t = np.arange(0.0, 5.0, 0.2)
4 plt.plot(t,t,'r--', t,t**2, 'bs', t,t**3,'g^')
5 plt.show()
```



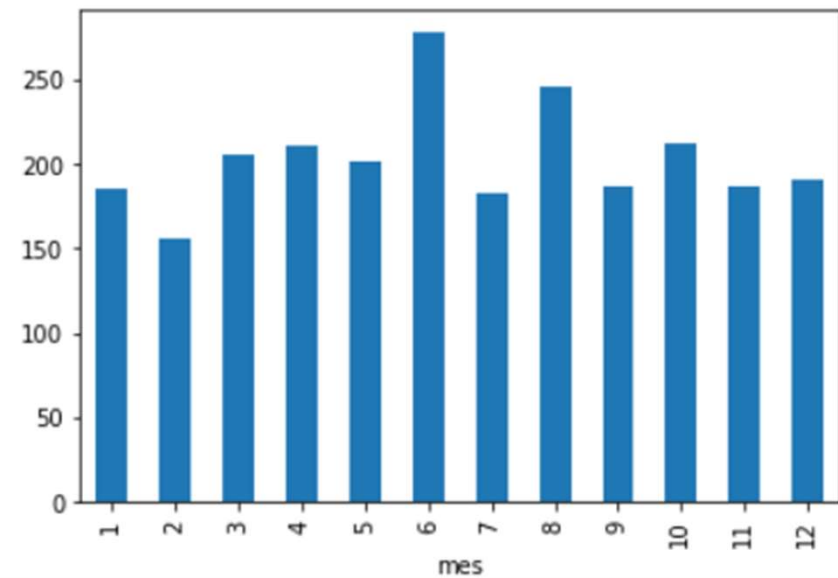
PANDAS

```
1 import numpy as np
2 import pandas as pd
3
4 df = pd.read_csv("./dat/obser2021.csv", sep=";")
5 df
6
7 df[df['obser'] == 'BCN']
8 df.loc[df.obser == 'BCN'].loc[df.component == 'N'].loc[df.plujamm > 8]
9
```

	anyy	mes	dia	obser	ventkh	component	plujamm	tmax	tmin	tmitjana
410	2011	7	26	BCN	146	N	9	1.86	-4.45	3.16
614	1985	9	19	BCN	20	N	10	36.34	4.67	15.84
998	1972	6	5	BCN	128	N	10	21.81	-6.01	13.91
1471	1999	6	9	BCN	28	N	9	42.99	-0.90	21.95
2013	1996	7	11	BCN	129	N	9	34.25	17.51	8.37
2089	2007	12	29	BCN	29	N	10	42.08	14.60	13.74
3087	2012	7	9	BCN	130	N	10	40.09	-4.66	22.38
3170	2008	8	22	BCN	122	N	9	25.96	-7.81	16.89
3651	1956	3	12	BCN	50	N	9	27.63	18.88	4.38
4217	1986	10	12	BCN	14	N	9	15.45	0.82	7.31

```
1 plot_data = df[df['obser'] == 'BCN']
2 plot_data = plot_data.groupby('mes')['plujamm'].sum()
3 plot_data.plot(kind='bar')
4
```

<AxesSubplot:xlabel='mes'>



TRATAMIENTO DE ERRORES

```
1 def agregar_una_vez (lista, el):
2
3     try:
4         if el not in lista:
5             lista.append (el)
6         else:
7             raise ValueError
8     except ValueError:
9         print ('Error: Imposible añadir elementos duplicados', el)
10    except Exception as e:
11        print (type(e), e)
12
13
14 #-----..Inicio de programa
15 elementos = [1, 5, -2]
16
17 agregar_una_vez (elementos, 10)
18 agregar_una_vez (elementos, 1)
19 agregar_una_vez (elementos, 'Hola')
```

Error: Imposible añadir elementos duplicados 1

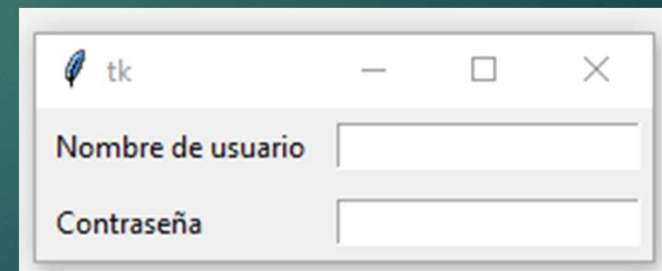
CLASES

```
1 class Pelicula:
2     def __init__(self, titulo, duracion = 0, lanzamiento = 1920):
3         self.titulo = titulo
4         self.duracion = duracion
5         self.lanzamiento = lanzamiento
6
7
8     def __str__(self):
9         cadena = f'Titulo: {self.titulo} dura : {self.duracion} lanzam: {self.lanzamiento}'
10        #cadena = 'Titulo: ' + str(self.titulo)
11        return cadena
12 #-----inicio
13
14 p1 = Pelicula ('1917', 180, 2020)
15 print(p1)
16 p2 = Pelicula ('Batman')
17 print(p2)
18
19
```

Titulo: 1917 dura : 180 lanzam: 2020
Titulo: Batman dura : 0 lanzam: 1920

TKINTER

```
1 from tkinter import *
2
3 root = Tk()
4
5 label = Label(root, text="Nombre de usuario")
6 label.grid(row=0, column=0, sticky="w", padx=5, pady=5)
7
8 entry = Entry(root)
9 entry.grid(row=0, column=1, padx=5, pady=5)
10 entry.config(justify="right", state="normal")
11
12 label2 = Label(root, text="Contraseña")
13 label2.grid(row=1, column=0, sticky="w", padx=5, pady=5)
14
15 entry2 = Entry(root)
16 entry2.grid(row=1, column=1, padx=5, pady=5)
17 entry2.config(justify="center", show="?")
18
19 root.mainloop()
20
```



BBDD

```
1 import sqlite3
2
3 conexion = sqlite3.connect('ejemplo.db')
4 cursor = conexion.cursor()
5
6 # Recuperamos los registros de la tabla de usuarios
7 cursor.execute("SELECT * FROM usuarios")
8
9 # Recorremos todos los registros con fetchall
10 # y los volcamos en una lista de usuarios
11 usuarios = cursor.fetchall()
12
13 # Ahora podemos recorrer todos los usuarios
14 for usuario in usuarios:
15     print(usuario)
16
17 conexion.close()
18
```

```
('Hector', 27, 'hector@ejemplo.com')
('Mario', 51, 'mario@ejemplo.com')
('Mercedes', 38, 'mercedes@ejemplo.com')
('Juan', 19, 'juan@ejemplo.com')
```

WEB con Json

```
1 import requests
2 import json
3 response = requests.get('https://gorest.co.in/public/v2/users')
4 # print(response.json())
5 for data in response.json():
6     print(data)
```

```
{'id': 2568, 'name': 'Rajan Joshi', 'email': 'rajan_joshi@wisoky.io', 'gender': 'male', 'status': 'active'}
{'id': 2567, 'name': 'Prathamesh Patel IV', 'email': 'patel_iv_prathamesh@haley.com', 'gender': 'female', 'status': 'inactive'}
{'id': 2566, 'name': 'Shwet Mehra', 'email': 'shwet_mehra@mcclure-marvin.biz', 'gender': 'male', 'status': 'inactive'}
{'id': 2565, 'name': 'Msgr. Shwet Kaul', 'email': 'shwet_msgr_kaul@damore.biz', 'gender': 'female', 'status': 'active'}
{'id': 2564, 'name': 'Akula Bhattacharya', 'email': 'akula_bhattacharya@wiegand-wilkinson.name', 'gender': 'female', 'status': 'active'}
{'id': 2563, 'name': 'Bhaaswar Mahajan', 'email': 'bhaaswar_mahajan@stracke.info', 'gender': 'female', 'status': 'active'}
{'id': 2562, 'name': 'Smriti Gill', 'email': 'gill_smriti@mosciski.biz', 'gender': 'male', 'status': 'inactive'}
{'id': 2561, 'name': 'Darshwana Nayar', 'email': 'nayar_darshwana@connelly.co', 'gender': 'male', 'status': 'active'}
{'id': 2560, 'name': 'Sitara Kaniyar IV', 'email': 'iv_sitara_kaniyar@zemlak-roob.biz', 'gender': 'female', 'status': 'inactive'}
{'id': 2559, 'name': 'Eekalabya Johar', 'email': 'johar_eekalabya@bernhard.net', 'gender': 'female', 'status': 'inactive'}
{'id': 2558, 'name': 'Krishnadasa Panicker', 'email': 'krishnadasa_panicker@koepp.co', 'gender': 'female', 'status': 'inactive'}
{'id': 2557, 'name': 'Mandakini Khatri VM', 'email': 'khatri_mandakini_vm@zemlak.name', 'gender': 'female', 'status': 'inactive'}
{'id': 2555, 'name': 'Sweta Ahluwalia', 'email': 'sweta_ahluwalia@reichert-kohler.net', 'gender': 'male', 'status': 'inactive'}
{'id': 2554, 'name': 'Devika Mehrotra', 'email': 'devika_mehrotra@smith.co', 'gender': 'male', 'status': 'active'}
{'id': 2553, 'name': 'Bharat Nair', 'email': 'bharat_nair@rutherford.co', 'gender': 'male', 'status': 'inactive'}
{'id': 2552, 'name': 'Bhavani Trivedi', 'email': 'bhavani_trivedi@robel.org', 'gender': 'male', 'status': 'active'}
{'id': 2551, 'name': 'Ganak Saini', 'email': 'ganak_saini@mitchell-paucek.co', 'gender': 'female', 'status': 'active'}
{'id': 2550, 'name': 'Suresh Varman', 'email': 'varman_suresh@conn.net', 'gender': 'female', 'status': 'inactive'}
{'id': 2549, 'name': 'Suresh Asan', 'email': 'asan_suresh@rau-medhurst.com', 'gender': 'male', 'status': 'active'}
{'id': 2548, 'name': 'Ananta Naik Sr.', 'email': 'sr_ananta_naik@mitchell.co', 'gender': 'male', 'status': 'active'}
```