# Programación Python

Dmitri Sidorov

# Ejemplos de uso de listas en Python

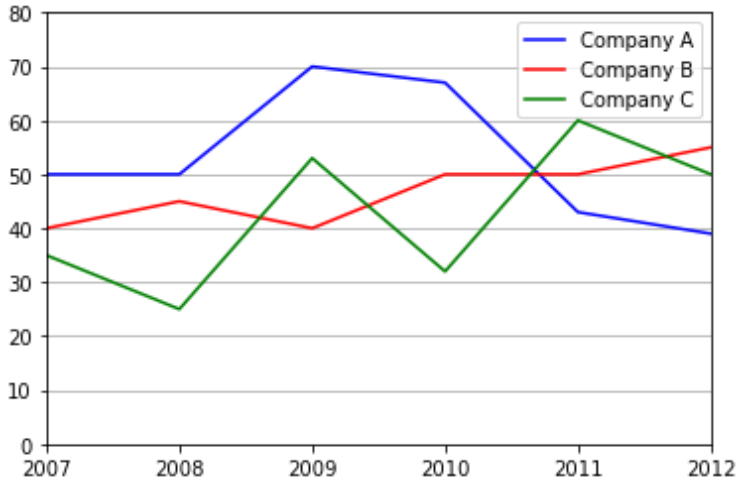
El código es un ejemplo de un juego de tres en raya usando listas para guarda r los movimientos de los jugadores:

```
J |   | O
  | J | O
  |   |
linea 1-33
columna 1-23
J |   | O
  | J | O
  |   | J
```

```
1. game =[list("J O"),list(" JO"),list("   ")]
2.
3. def printau (f1) :
4.     print (f1[0][0], "|", f1[0][1], "|", f1[0][2])
5.     print (f1[1][0], "|", f1[1][1], "|", f1[1][2])
6.     print (f1[2][0], "|", f1[2][1], "|", f1[2][2])
7.
8. printau(game)
9. linea = int(input("linea 1-3"))
10.     columna = int(input("columna 1-2"))
11.
12.     if game[linea-1][columna-1] != " ":
13.         print("casilla ocupada")
14.     else:
15.         game[linea-1][columna-1] = "J"
16.
17.     printau(game)
```

# Uso de las librerías Numpy y Matplot para generar graficos

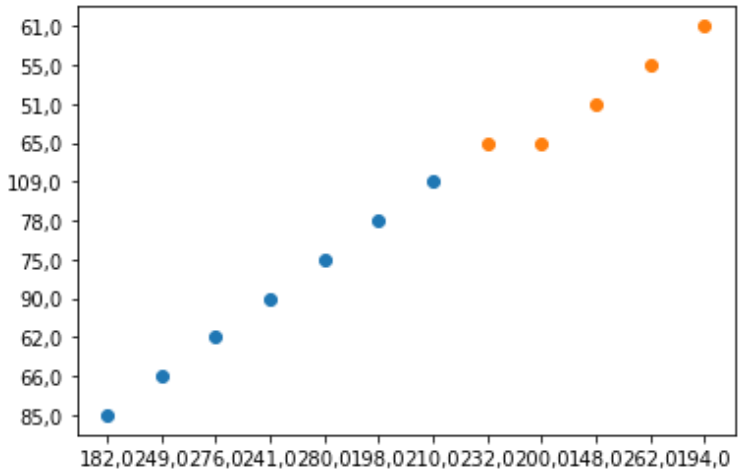
El código parte de 3 listas que se usan para g enerar un gráfico de líneas usando la librería Matplotlib:




```
1. import numpy as np
2. import matplotlib.pyplot as plt
3.
4. A = [50, 50, 70, 67, 43, 39]
5. B = [40, 45, 40, 50, 50, 55]
6. C = [35, 25, 53, 32, 60, 50]
7.
8. X = np.arange(2007, 2013)
9. X = X.tolist()
10.
11.     plt.plot(X, A, label="Company A", color="blue")
12.     plt.plot(X, B, label="Company B", color="red")
13.     plt.plot(X, C, label="Company C", color="green")
14.     plt.axis([2007, 2012, 0, 80])
15.     plt.grid(axis="y")
16.     plt.legend()
17.     plt.show()
```

# Uso de las librerías Pandas, Numpy y Matplot.

El código parte de un archivo csv externo que tratamos con la librería Pandas para manejarlo como DataFrame, el cual se usa para generar un gráfico de puntos usando la librería Matplotlib:




```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read csv('dat/colesterol.csv', sep=';')

#reemplaza valores null por 0
df = df.dropna()

    #ordena por nivel de colesterol mas alto
    print(df.sort_values(by='colesterol', ascending=False))

    #creando dataframes para cada sexo
    dfh = df[df['sexo']== 'H']
    dfm = df[df['sexo']== 'M']

    #creando dataframe sin nombres y apellidos
    dfa = df.drop(['nombre'], axis=1)
    #guardamos los datos anónimos en un archivo csv
    dfa.to_csv(r'dat/datos_anonimos.csv', index=False)


    plt.scatter(dfh.colesterol, dfh.peso)
    plt.scatter(dfm.colesterol, dfm.peso)
```

# Programación orientada a objetos.

El código utiliza el paradigma de objetos par a maneja películas y catálogos de películas utilizando sus métodos y atributos.

```
1. class Pelicula:
2.     def __init__(self, titulo, duracion, lanzamiento):
3.         self.titulo=titulo
4.         self.duracion=duracion
5.         self.lanzamiento=lanzamiento
6.     def __str__(self):
7.         cadena = self.titulo+', año: '+str(self.lanzamiento)
8.         return cadena
9.
10.      class Catalogo:
11.          def __init__(self, lista=[]):
12.              self.peliculas = lista
13.          def agregar(self, pelicula):
14.              if pelicula not in self.peliculas:
15.                  self.peliculas.append(pelicula)
16.              else:
17.                  print(pelicula.titulo+' ya existe')
18.          def mostrar(self):
19.              for pelicula in self.peliculas:
20.                  print(pelicula)
21.          def vaciar(self):
22.              self.peliculas = []
23.          def borrar(self, titulo):
24.              for p in self.peliculas:
25.                  if p.titulo == titulo:
26.                      self.peliculas.remove(p)
27.                      break
28.
29.      p1 = Pelicula('Dune', 90, 2021)
30.      p2 = Pelicula('The Thing', 80, 1982)
31.      p3 = Pelicula('99 Francs', 86, 2007)
32.
33.      catalogo = Catalogo()
34.
35.      catalogo.agregar(p1)
36.      catalogo.agregar(p2)
37.      catalogo.agregar(p2)
38.      catalogo.agregar(p3)
39.      catalogo.agregar(Pelicula('Braveheart',100,1999))
40.
41.      print(catalogo.mostrar())
42.
43.      catalogo.borrar('Dune')
44.      print(catalogo.mostrar())
```

# Captura y tratamiento de errores

Ejemplo básico de captura de errores y detección de errores específicos.
El código recoge el error "ZeroDivisionError" y devuelve un mensaje personalizado.

```
1. try:
2.     n = float(input('introducir numero: '))
3.     5/n
4.
5. except ValueError :
6.     print("ha introducido una cadena")
7.
8. except ZeroDivisionError :
9.     print("no se puede dividir por 0")
10.
11.     except Exception as e:
12.         print(type(e))
13.         print("hay un error")
```