



PROGRAMACION CON PYTHON

SIMONE MORO

ABRIL 2022

100

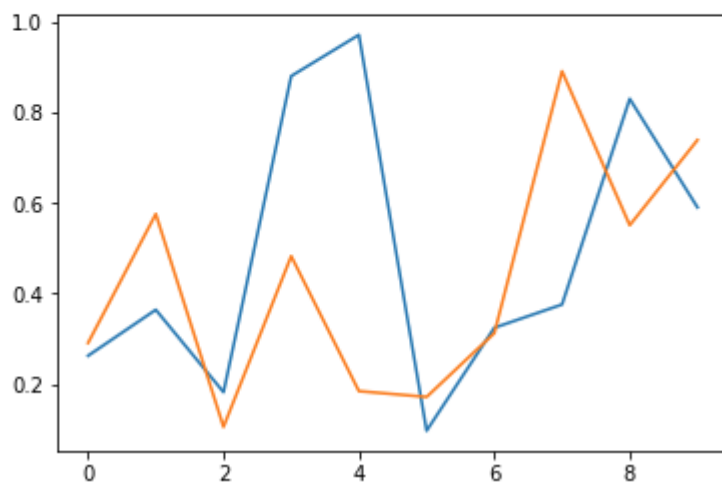
```
In [ ]: 1 import numpy as np
2 import pandas as pd
3
4 plt.figure(figsize=(10,8))
5 plt.imshow(word_cloud)
6 plt.axis('off')
7 plt.tight_layout(pad=0)
8 plt.show()
9
10 word_cloud = WordCloud(height=800, width=800, background_color='white',max_words=150, min_font_size=10)
11
12 plt.figure(figsize=(10,8))
13 plt.imshow(word_cloud)
14 plt.axis('off')
15 plt.tight_layout(pad=0)
16 plt.show()
```



Graficos con MATPLOTLIB Y numpy

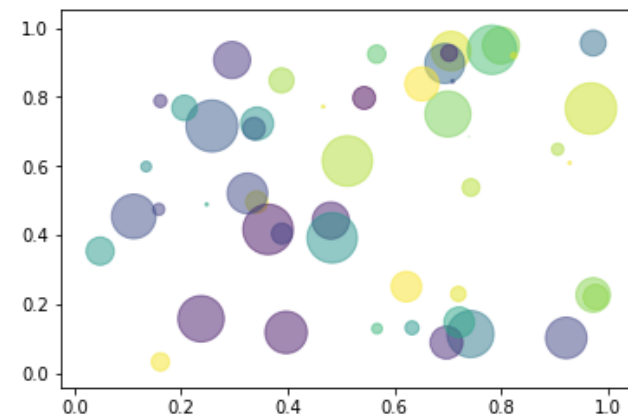
In [1]:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 plt.plot(np.random.rand(10))
4 plt.plot(np.random.rand(10))
5 plt.show()
```



In [5]:

```
1 # EJEMPLO DE GRAFICO DE BURBUJAS
2
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 # Fixing random state for reproducibility
7 np.random.seed(19680801)
8
9
10 N = 50
11 x = np.random.rand(N)
12 y = np.random.rand(N)
13 colors = np.random.rand(N)
14 area = (30 * np.random.rand(N))**2 # 0 to 15 point radii
15
16 plt.scatter(x, y, s=area, c=colors, alpha=0.5)
17 plt.show()
```



Graficos estadísticos con MATHPLOTLIB y PLOTLY.EXPRESS

```
import matplotlib.pyplot as plt
|
# definitions for the axes
left, width = 0.1, 0.65
bottom, height = 0.1, 0.65
spacing = 0.005

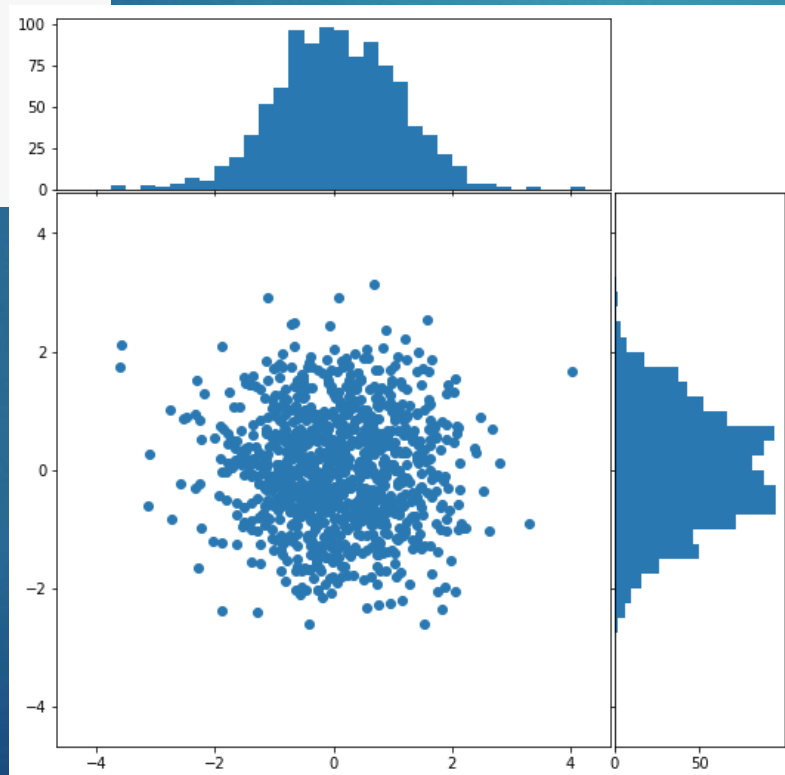
rect_scatter = [left, bottom, width, height]
rect_histx = [left, bottom + height + spacing, width, 0.2]
rect_histy = [left + width + spacing, bottom, 0.2, height]

# start with a square Figure
fig = plt.figure(figsize=(8, 8))

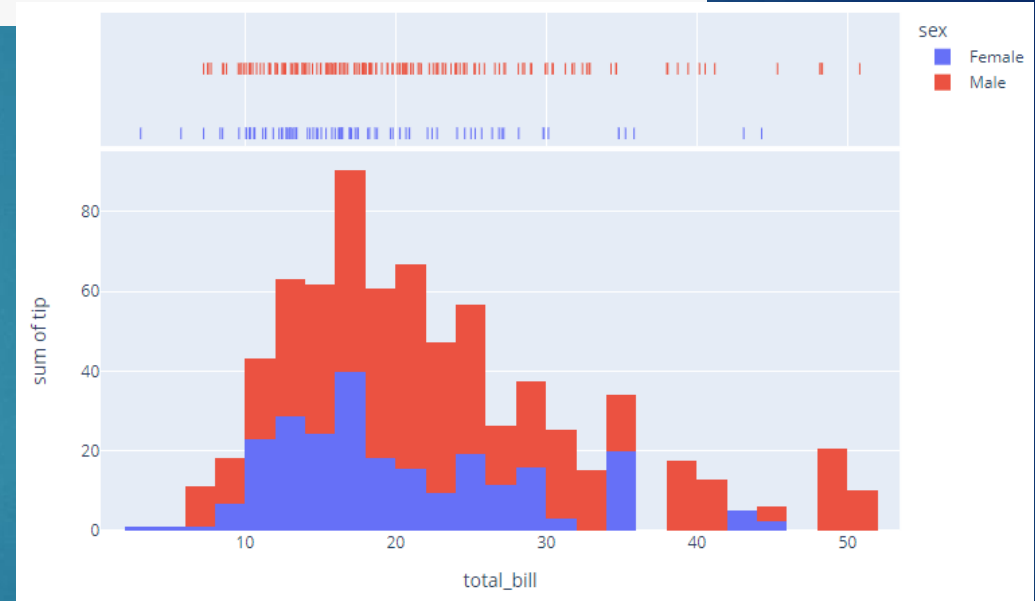
ax = fig.add_axes(rect_scatter)
ax_histx = fig.add_axes(rect_histx, sharex=ax)
ax_histy = fig.add_axes(rect_histy, sharey=ax)

# use the previously defined function
scatter_hist(x, y, ax, ax_histx, ax_histy)

plt.show()
```

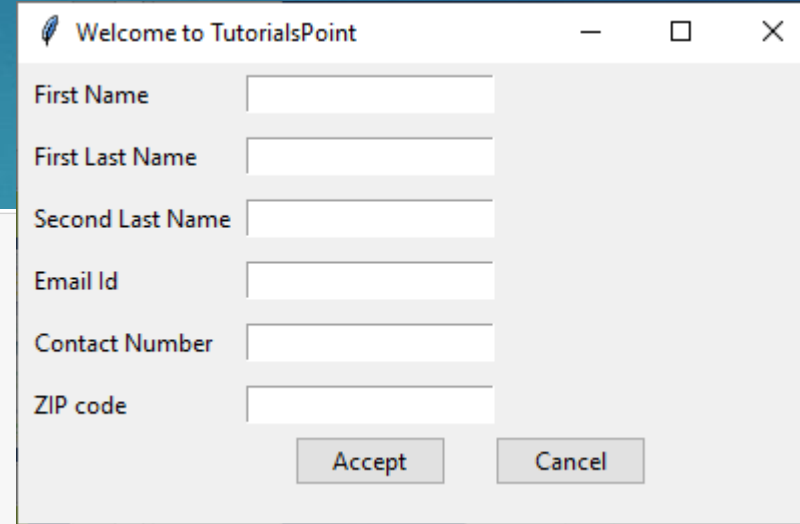


```
import plotly.express as px
df = px.data.tips()
fig = px.histogram(df, x="total_bill", y="tip", color="sex", marginal="rug",
                  hover_data=df.columns)
fig.show()
```



Creacion de formularios con tkinter

```
In [1]: 1 #----- PRACTICA
2 # -*- coding: utf-8 -*-
3 """
4 Created on Thu Sep 10 19:00:21 2020
5
6 @author: Usuario
7 """
8
9 from tkinter import *
10 from tkinter import ttk
11
12 window = Tk()
13 window.title("Welcome to Tutorialspoint")
14 window.geometry('400x200')
15 window.minsize(width=400, height=230)
16
17 frame = Frame(window)
18 Label(frame ,text = "First Name").grid(row = 0,column = 0, sticky="w", padx=5, pady=5)
19 Label(frame ,text = "First Last Name").grid(row = 1,column = 0, sticky="w", padx=5, pady=5)
20 Label(frame ,text = "Second Last Name").grid(row = 2,column = 0, sticky="w", padx=5, pady=5)
21 Label(frame ,text = "Email Id").grid(row = 3,column = 0, sticky="w", padx=5, pady=5)
22 Label(frame ,text = "Contact Number").grid(row = 4,column = 0, sticky="w", padx=5, pady=5)
23 Label(frame ,text = "ZIP code").grid(row = 5,column = 0, sticky="w", padx=5, pady=5)
24
25 Entry(frame).grid(row = 0,column = 1)
26 Entry(frame).grid(row = 1,column = 1)
27 Entry(frame).grid(row = 2,column = 1)
28 Entry(frame).grid(row = 3,column = 1)
29 Entry(frame).grid(row = 4,column = 1)
30 Entry(frame).grid(row = 5,column = 1)
31
32 ttk.Button(frame ,text="Accept").grid(row=6,column=1)
33 ttk.Button(frame ,text="Cancel").grid(row=6,column=2)
34 frame.pack(anchor=NW, expand=1)
35
36 window.mainloop()
37
```



The screenshot shows a Tkinter window titled "Welcome to Tutorialspoint" with a standard macOS-style title bar (minimize, maximize, close buttons). The window contains a form with the following elements:

- First Name
- First Last Name
- Second Last Name
- Email Id
- Contact Number
- ZIP code

At the bottom of the form are two buttons: "Accept" and "Cancel".

Crear juego con OOP

```
class Coche(object):
    """ Coche, se desplaza derecha e izquierda. """

    def __init__(self, juego):
        self.ancho = 137 # Ancho del coche
        self.x_acelera = 0 # parámetros para desplazamiento
        self.velocidad = 0 # y velocidad

        self.x = juego.ancho * 0.45 # parámetros de posición
        self.y = juego.alto * 0.8
        self.img = pygame.image.load('.\\img\\yellowcar.png')

    def dibuja_y_mueve (self, juego):
        juego.ventana.fill(WHITE) # Blanquea la pantalla
        self.x += self.x_acelera # recalcula la coordenada x
        juego.ventana.blit(self.img, (self.x, self.y)) # Dibuja el coche

    def acelerar (self, valor):
        self.x_acelera = valor

    def calcula_colision (self, caja):
        if self.y < (caja.y + caja.alto):
            if (self.x > caja.x and self.x < (caja.x + caja.ancho)) or \
                ((self.x + self.ancho > caja.x) and (self.x + self.ancho) < (caja.x + caja.ancho)):
                return True
        return False

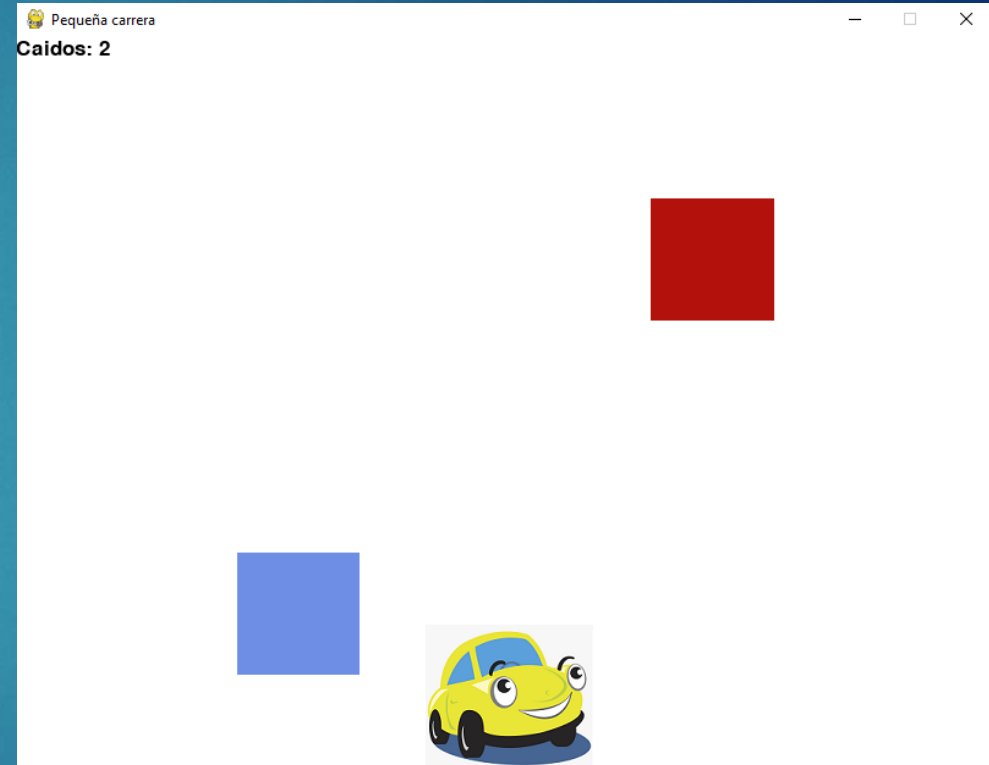
class Caja(object):
    """ Caja, cae vertical """

    def __init__(self, juego, color, pos_y):
        self.ancho = 100
        self.alto = 100
        self.speed = 3
        self.x = random.randrange(0, juego.ancho)
        self.y = pos_y
        self.color = color
        self.dibuja (juego)

    def dibuja (self, juego):
        pygame.draw.rect(juego.ventana, self.color, [self.x, self.y, self.ancho, self.alto])

    def mueve (self, juego):
        #----- baja la caja
        self.y += self.speed

        # Si la caja sale por abajo sin choque reaparece por arriba
        if self.y > juego.alto:
            self.y = 0 - self.alto # Si la caja se sale de la pantalla
            self.x = random.randrange(0, juego.ancho) # se vuelve a situar arriba en otra
                                                    # coordenada aleatoria
            juego.numCaídas += 1 # se suma 1 a las cajas caídas
            self.speed += 0.5 #1 # se aumenta la velocidad
            self.ancho += (juego.numCaídas * 0.5) # 1.2 # va aumentando la anchura de la caja
```



Hacer una request y visualizarla con JSON

```
import requests
import json
response = requests.get('https://gorest.co.in/public/v2/users')
# print(response.json())
for data in response.json():
    print(data['name'], '\t\t', data['email'])
```

Rajan Joshi	rajan_joshi@wisoky.io
Prathamesh Patel IV	patel_iv_prathamesh@haley.com
Shwet Mehra	shwet_mehra@mcclure-marvin.biz
Msgr. Shwet Kaul	shwet_msgr_kaul@damore.biz
Akula Bhattacharya	akula_bhattacharya@wiegand-wilkinson.name
Bhaaswar Mahajan	bhaaswar_mahajan@stracke.info
Smriti Gill	gill_smriti@mosciski.biz
Darshwana Nayar	nayar_darshwana@connelly.co
Sitara Kaniyar IV	iv_sitara_kaniyar@zemplak-roob.biz
Eekalabya Johar	johar_eekalabya@bernhard.net
Krishnadasa Panicker	krishnadasa_panicker@koepp.co
Mandakini Khatri VM	khatri_mandakini_vm@zemplak.name
Sweta Ahluwalia	sweta_ahluwalia@reichert-kohler.net
Devika Mehrotra	devika_mehrotra@smith.co
Bharat Nair	bharat_nair@rutherford.co
Bhavani Trivedi	bhavani_trivedi@robel.org
Ganak Saini	ganak_saini@mittchell-paucek.co
Suresh Varman	varman_suresh@conn.net
Suresh Asan	asan_suresh@rau-medhurst.com
Ananta Naik Sr.	sr_ananta_naik@mittchell.co