Trombini_Quentin_Week4

S04_01

S04_01_a

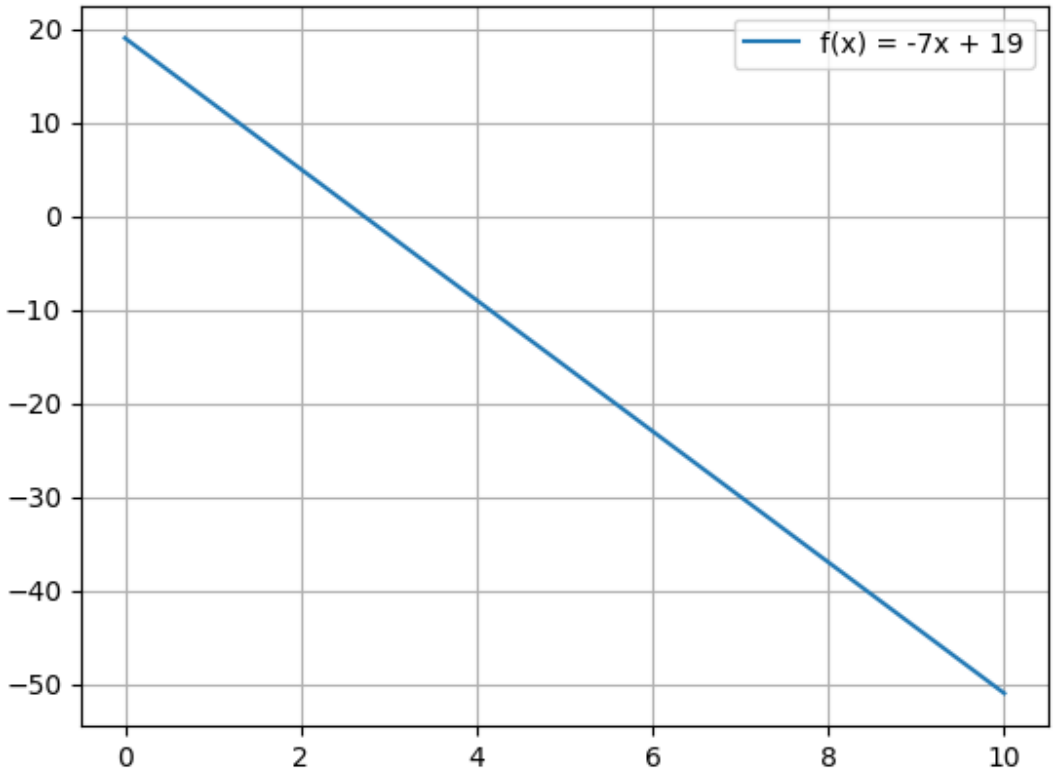
```
import matplotlib.figure
import matplotlib.backends
import numpy as np

x = np.linspace(0,10,1000)
y = (-7 * x) + 19

fig = matplotlib.figure.Figure()
ax = fig.add_subplot(111)
ax.plot(x,y,label = "f(x) = -7x + 19")
ax.legend()
ax.grid()


fig.savefig("1.png")
```

S04_01_b



S04_01_c

first of all we import all the useful library and define an alias for numpy.

then we will define x and y, x will be a linear array of value and y will be a function.

using **ax.plot()** we will draw this function using the value of each x for each y.

then we will use function to enhance our output and make it more readeable for example **.grid will add a grid behind the graph**

S04_02

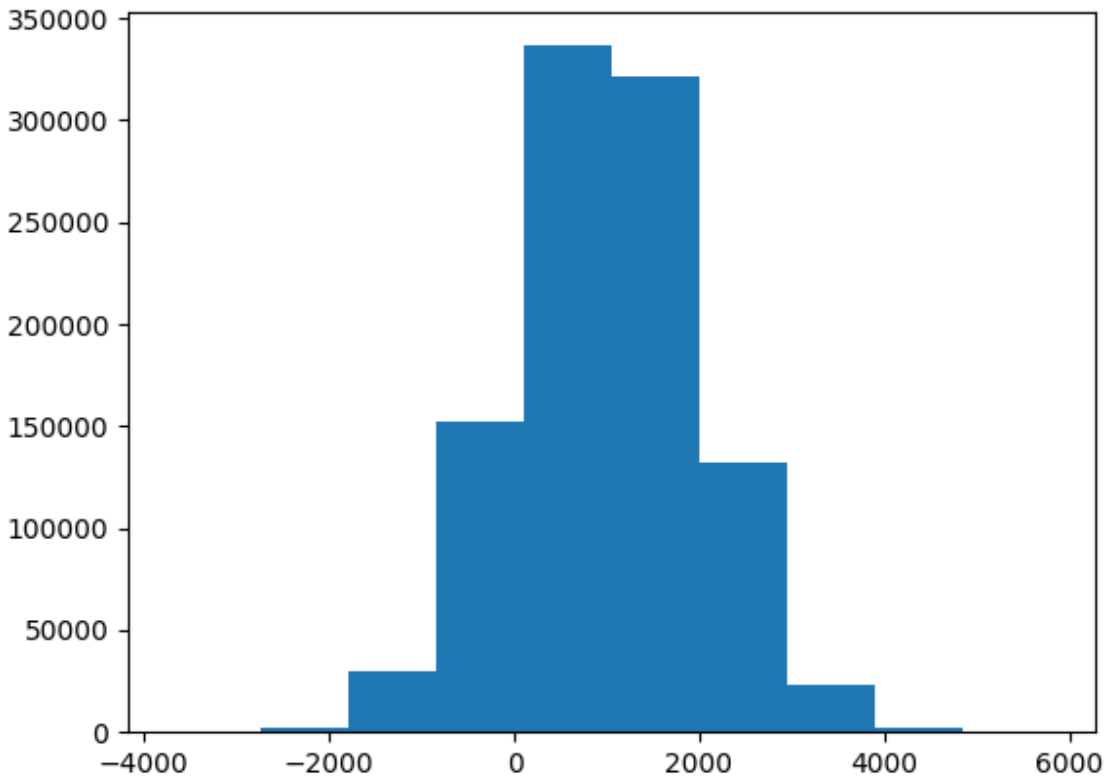
S04_02_a

```
import numpy as np
import matplotlib.pyplot as plt

rng = np.random.Generator(np.random.PCG64())

array = rng.normal(1000,1000,10**6)
plt.hist(array)
plt.savefig("2.png")
```

S04_02_b



S04_02_c

I start by importing the two useful library and define alias for them

Then I use **np.random** and **rng.normal** to get a gaussian distribution of the values the we use matplotlib to transform the array into an historgram

---

S04_03

S04_03_a

```
import numpy as np
import matplotlib.backends.backend_agg
import matplotlib.figure as fig
import matplotlib.animation as animation

theta = np.radians(30)
g = 9.81
v = 30
total_time = (2 * v * np.sin(theta)) / g

nb_frame = 60
frames = []
figure = fig.Figure()
canvas = matplotlib.backends.backend_agg.FigureCanvasAgg(figure)
ax = figure.add_subplot(111)

for i in range(nb_frame):
    ax.set_xlim(0,100)
    ax.set_ylim(0,30)

    t = (i/(nb_frame - 1))*total_time
    x = v*t*np.cos(theta)
    y = v*t*np.sin(theta) - (0.5*g*t**2)
    point, = ax.plot(x,y,"bo")

    ax.set_aspect("equal")
    frames.append([point])

anim = animation.ArtistAnimation(figure,frames,interval = 50)
anim.save("3.mp4",dpi = 225)
```

S04_03_b

https://youtu.be/5D9_tsTpjGU

S04_03_c

First of all we initiate all the useful variable for the physics part of the algorithm,

Then we initialize all the animation-related variable

now we create a loop for each frame in our future animation

we set each frames axis to 30/100 to see the whole animation

then we will compute for each frame X and Y using the equation of a ball motion

then we add the created point to the list of frame

then we create an animation compiling all our frames and save it