

# numpydemo

December 27, 2022

```
[17]: import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import datetime as dt
import math
```

```
[28]: A=np.array([[1,2,3],[4,5,6]])
print(A)

Af=np.array([1,2,3],float)
print(Af)
```

```
[[1 2 3]
 [4 5 6]]
[1. 2. 3.]
0
1
2
3
```

```
[30]: a = np.array(42)
b = np.array([1, 2, 3, 4, 5])
c = np.array([[1, 2, 3], [4, 5, 6]])
d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])

print(a.ndim)
print(b.ndim)
print(c.ndim)
print(d.ndim)
print(d[0])
```

```
0
1
2
3
[[1 2 3]
 [4 5 6]]
```

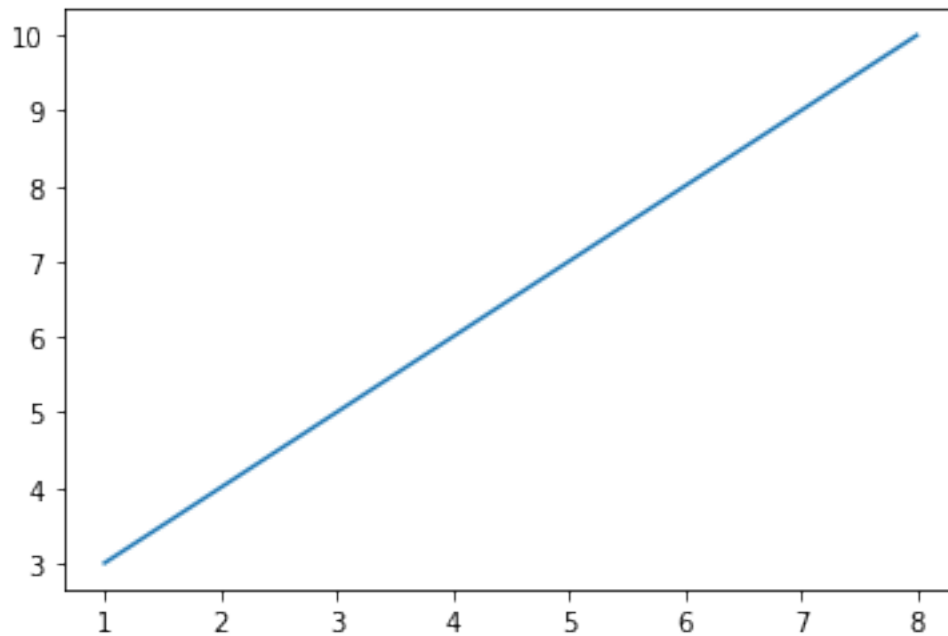
```
[27]: print(np.arange(0,1,0.2))
```

```
np.linspace(0,2*np.pi,10)
```

```
[0.  0.2 0.4 0.6 0.8]
```

```
[27]: array([0.          , 0.6981317 , 1.3962634 , 2.0943951 , 2.7925268 ,  
          3.4906585 , 4.1887902 , 4.88692191, 5.58505361, 6.28318531])
```

```
[4]: xpoints = np.array([1, 8])  
ypoints = np.array([3, 10])  
  
plt.plot(xpoints, ypoints)  
plt.show()
```



```
[5]: x=[1,2,3,4,5,6,7,8,9]  
y=[1,2,3,4,5,6,7,8,9]  
xarr=np.array(x)  
yarr=np.array(y)  
x33=xarr.reshape(3,3)  
y33=yarr.reshape(3,3)  
print('list reshaped to array: \n')  
print(x33)  
print('product of the two matrices is : \n')  
product=np.matmul(x33,y33)  
print(product)
```

list reshaped to array:

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

product of the two matrices is :

```
[[ 30  36  42]
 [ 66  81  96]
 [102 126 150]]
```

```
[6]: arr = np.array([1, 2, 3, 4, 5, 4, 4])

x = np.where(arr == 4)

print(x)
```

```
(array([3, 5, 6], dtype=int64),)
```

```
[7]: spreadsheet = pd.read_csv('C:/Users/ANIRBAN/Downloads/archive/
    ↳Temperature_And_Precipitation_Cities_IN/Chennai_1990_2022_Madras.csv')
```

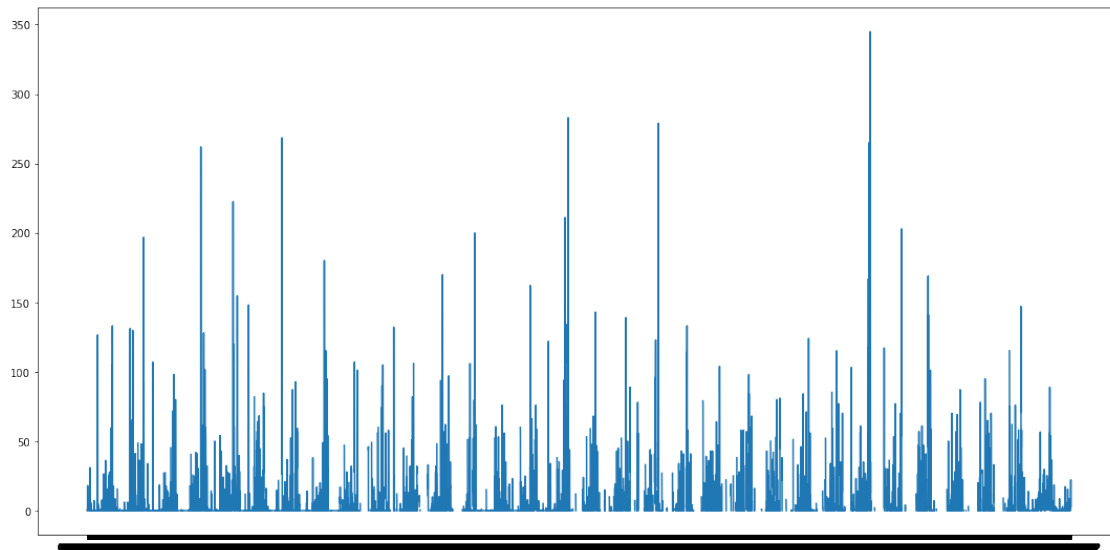
```
[8]: spreadsheet.head()
```

```
[8]:
```

	time	tavg	tmin	tmax	prcp
0	01-01-1990	25.2	22.8	28.4	0.5
1	02-01-1990	24.9	21.7	29.1	0.0
2	03-01-1990	25.6	21.4	29.8	0.0
3	04-01-1990	25.7	NaN	28.7	0.0
4	05-01-1990	25.5	20.7	28.4	0.0

```
[9]: plt.rcParams["figure.figsize"] = [15.00, 7.50]
plt.rcParams["figure.autolayout"] = True
columns = ["time", "prcp"]
arr=spreadsheet.values
print(arr)
plt.plot(spreadsheet.time, spreadsheet.prcp)
plt.show()
```

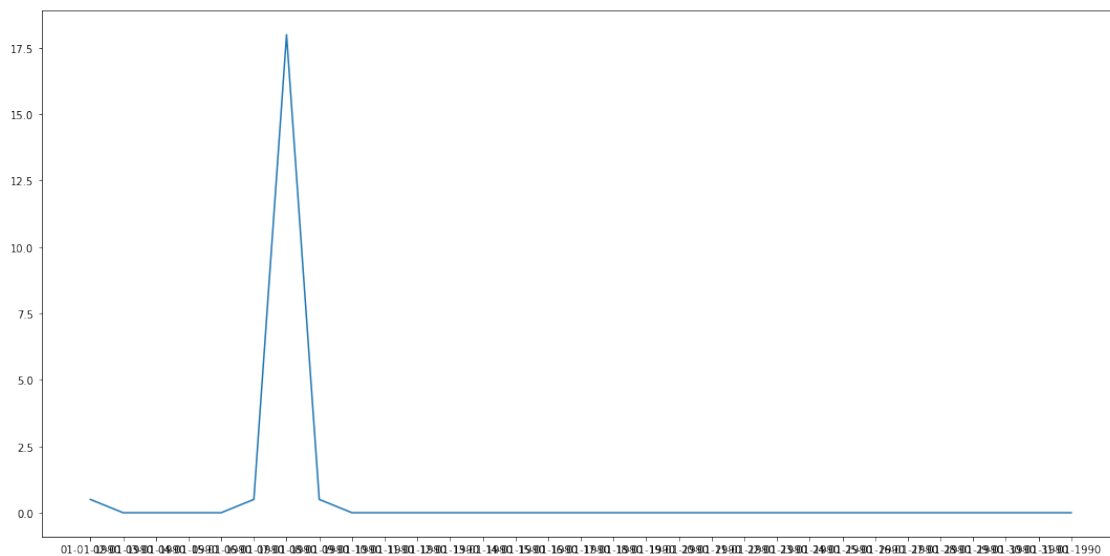
```
['01-01-1990' 25.2 22.8 28.4 0.5]
['02-01-1990' 24.9 21.7 29.1 0.0]
['03-01-1990' 25.6 21.4 29.8 0.0]
...
['23-07-2022' 27.4 24.7 32.6 18.6]
['24-07-2022' 27.8 25.0 33.3 9.1]
['25-07-2022' 28.1 25.4 32.6 2.9]]
```



```
[10]: nparr=np.array(arr)
```

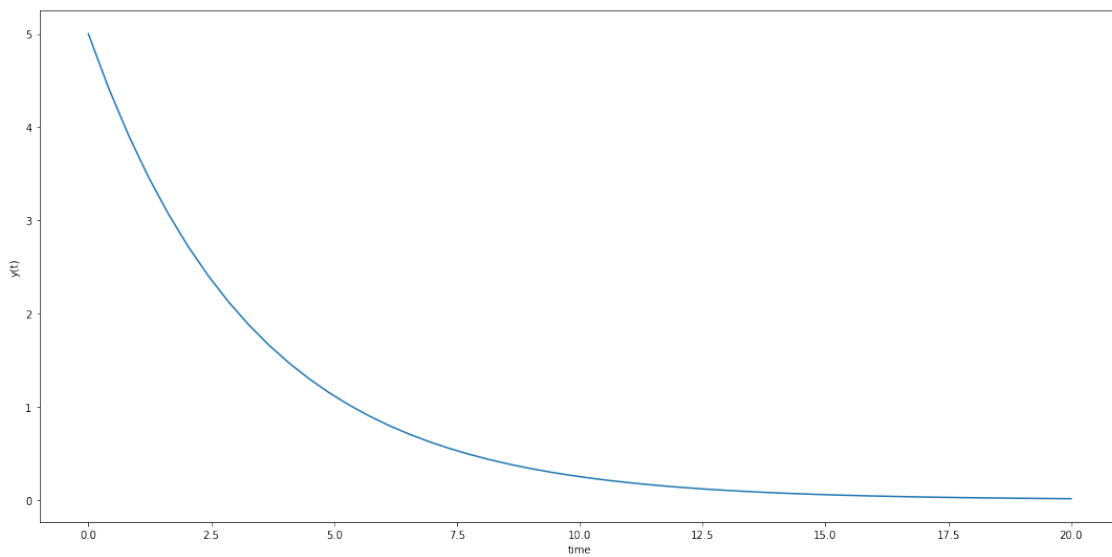
```
[11]: timedat=nparr[:,0]
prcpdat=nparr[:, 4]
print(prcpdat)
print(len(prcpdat))
newtime=timedat[0:31]
newprcp = prcpdat[0:31]
plt.plot(newtime, newprcp)
plt.show()
```

```
[0.5 0.0 0.0 ... 18.6 9.1 2.9]
11894
```



```
[12]: from scipy.integrate import odeint
```

```
def model(y,t):  
    k = 0.3  
    dydt = -k * y  
    return dydt  
  
# initial condition  
y0 = 5  
  
# time points  
t = np.linspace(0,20)  
  
# solve ODE  
y = odeint(model,y0,t)  
  
# plot results  
plt.plot(t,y)  
plt.xlabel('time')  
plt.ylabel('y(t)')  
plt.show()
```



```
[26]: from scipy.integrate import odeint
```

```

def model(y,t):
    g=9.8
    l=0.5
    k=g/l;
    return (y[1],-k*y[0])

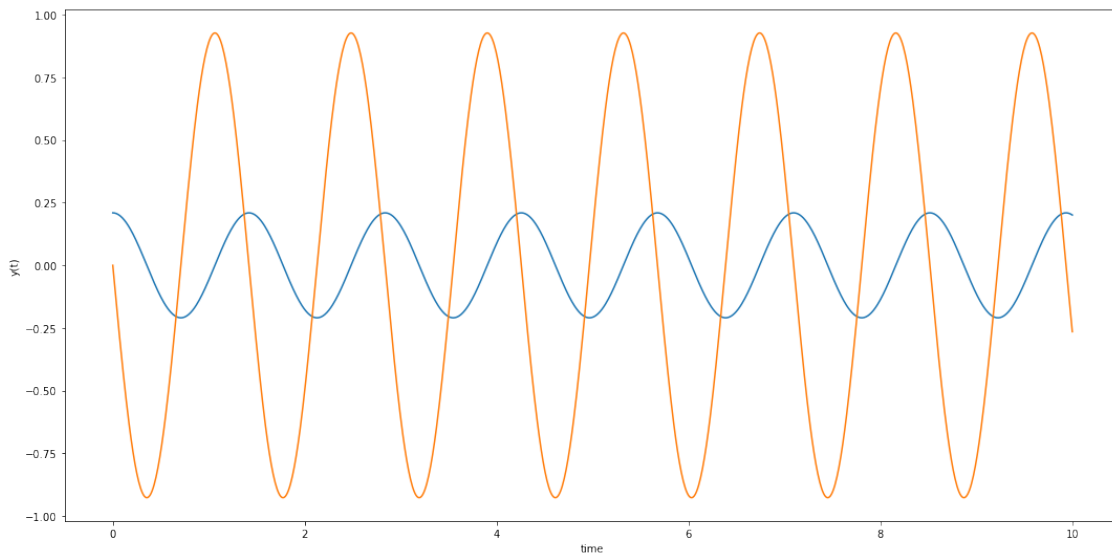
# initial condition
y0 = [(math.pi)/15,0]

# time points
t = np.linspace(0,10,1000)

# solve ODE
y = odeint(model,y0,t)

# plot results
plt.plot(t,y)
plt.xlabel('time')
plt.ylabel('y(t)')
plt.show()

```



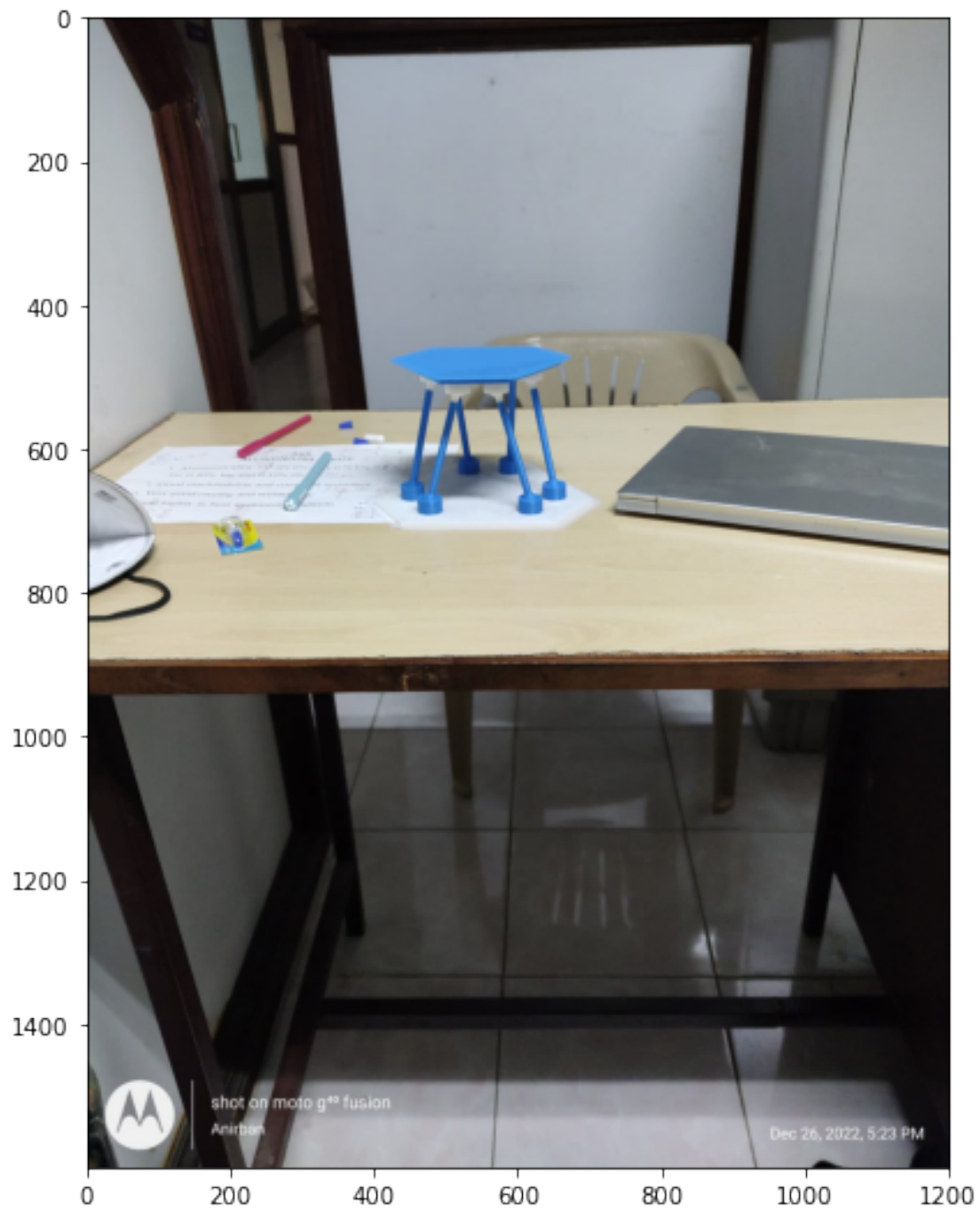
```

[39]: from scipy import misc
import imageio
f = misc.face()
imageio.imsave('face.png', f) # uses the Image module (PIL)
import matplotlib.pyplot as plt
plt.imshow(f)
plt.show()

```



```
[38]: from PIL import Image as PImage
      im = PImage.open('C:/Users/ANIRBAN/Downloads/spm_3dprint.jpeg')
      plt.imshow(im)
      plt.show()
```



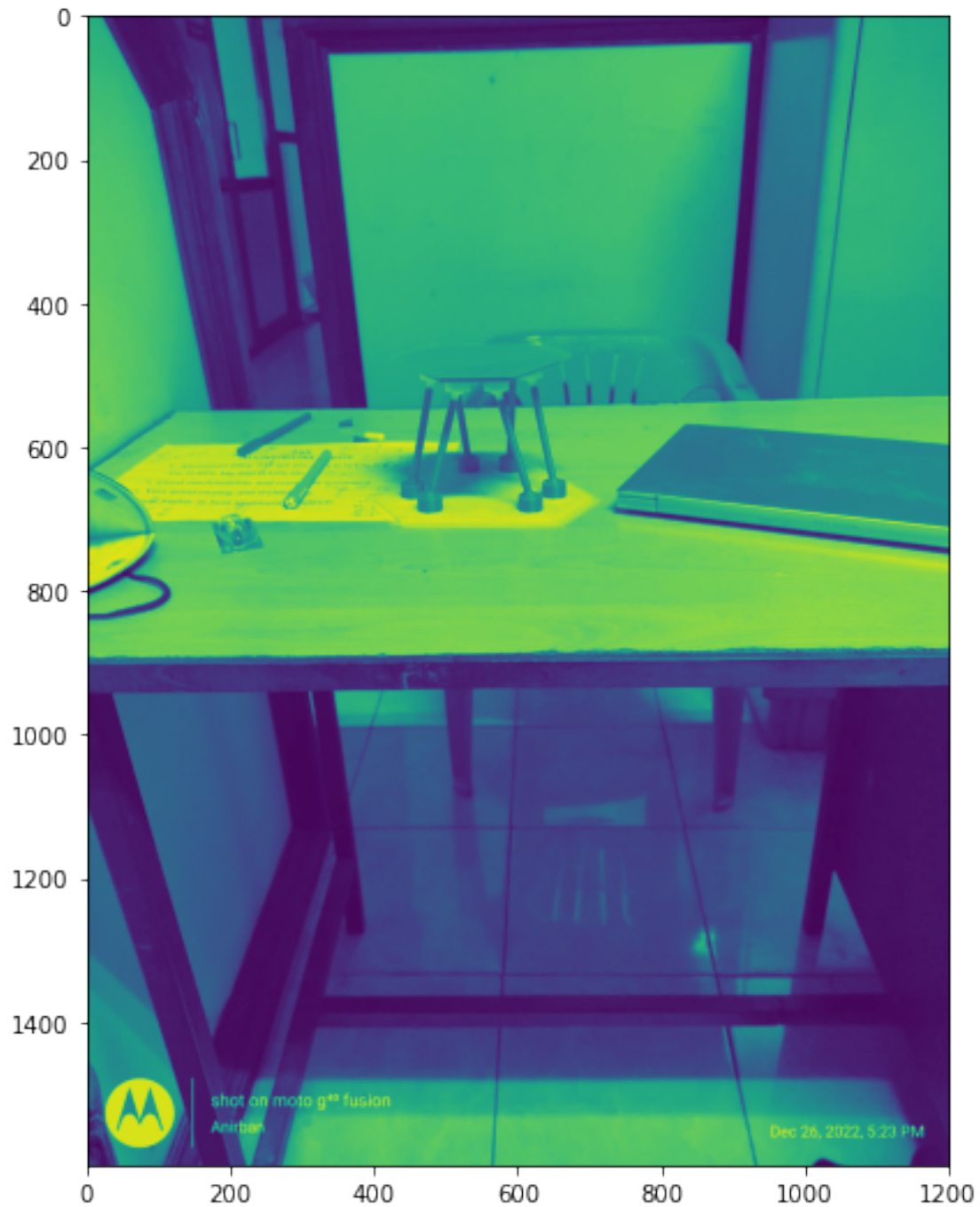
```
[43]: f = misc.face(gray=True)
      plt.imshow(f, cmap=plt.cm.gray)
```

```
[43]: <matplotlib.image.AxesImage at 0x2393d545eb0>
```

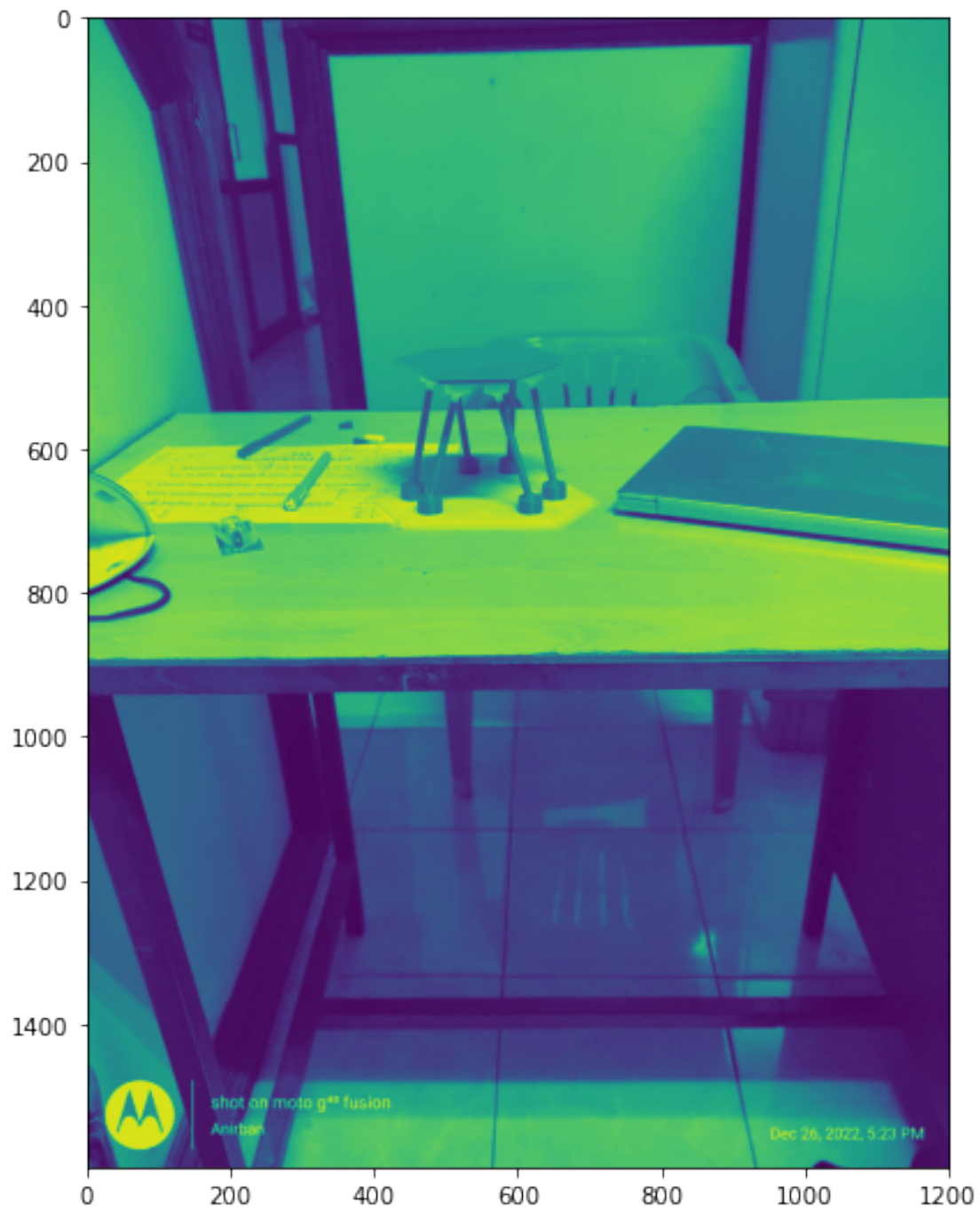




```
[49]: img = np.mean(im, axis=2)
      plt.imshow(img)
      plt.show()
```



```
[53]: from PIL import Image
img2 = Image.open('C:/Users/ANIRBAN/Downloads/spm_3dprint.jpeg').convert('L')
img2.save('pil-greyscale.png')
plt.imshow(img2)
plt.show()
```



```
[74]: face = misc.face(gray=True)
      print(face)
      len(face)
      print(face[0])
```

```
[[114 130 145 ... 119 129 137]
```

```
[ 83 104 123 ... 118 134 146]
[ 68  88 109 ... 119 134 145]
...
[ 98 103 116 ... 144 143 143]
[ 94 104 120 ... 143 142 142]
[ 94 106 119 ... 142 141 140]]
[114 130 145 ... 119 129 137]
```

[ ]: