
Animation

Tkinter를 사용

```
# https://infinitt.tistory.com/49?category=1076177
from matplotlib import pyplot as plt
from matplotlib import animation
import numpy as np
import random
import time
#
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from matplotlib.figure import Figure
import Tkinter as Tk
```

```
fig = plt.figure() #figure made
```

```
ax = plt.subplot(211, xlim=(0, 50), ylim=(0, 1024))
ax_2 = plt.subplot(212, xlim=(0, 50), ylim=(0, 512))
```

```
max_points = 50
max_points_2 = 50
```

```
line, = ax.plot(np.arange(max_points),
                np.ones(max_points, dtype=np.float)*np.nan, lw=1, c='blue',ms=1)
line_2, = ax_2.plot(np.arange(max_points_2),
                   np.ones(max_points, dtype=np.float)*np.nan, lw=1,ms=1)
```

```
def init():
    return line
def init_2():
    return line_2
```

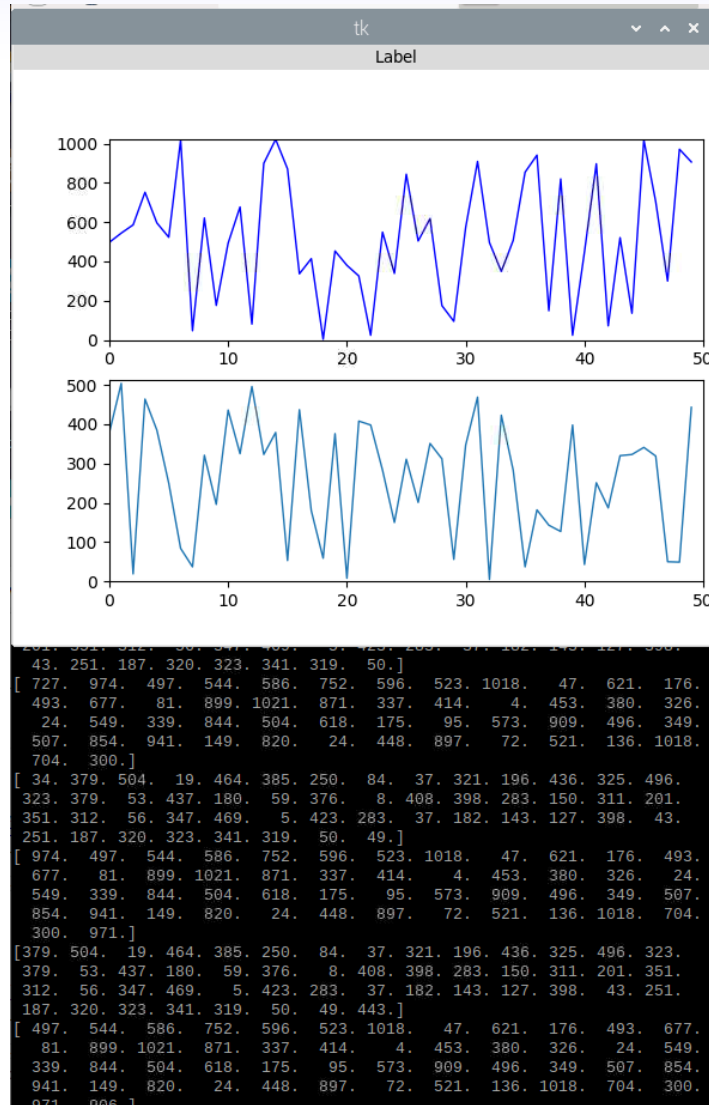
```
def animate(i):
    y = random.randint(0,1024)
    old_y = line.get_ydata()
    new_y = np.r_[old_y[1:], y]
    line.set_ydata(new_y)
    print(new_y)
    return line
```

```
def animate_2(i):
    y_2 = random.randint(0,512)
    old_y_2 = line_2.get_ydata()
    new_y_2 = np.r_[old_y_2[1:], y_2]
    line_2.set_ydata(new_y_2)
    print(new_y_2)
    return line_2
```

```
root = Tk.Tk() # Add
label = Tk.Label(root,text="Label").grid(column=0, row=0) # Added
canvas = FigureCanvasTkAgg(fig, master=root) #
canvas.get_tk_widget().grid(column=0,row=1) #
```

```
anim = animation.FuncAnimation(fig, animate , init_func= init ,frames=200, interval=50, blit=False)
anim_2 = animation.FuncAnimation(fig, animate_2 , init_func= init_2 ,frames=200, interval=10, blit=False)
Tk.mainloop()
```

결과



Sin함수

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

```
TWOPI = 2*np.pi
```

```
fig, ax = plt.subplots()
```

```
t = np.arange(0.0, TWOPI, 0.001)
```

```
s = np.sin(t)
```

```
l = plt.plot(t, s)
```

```
ax = plt.axis([0,TWOPI,-1,1])
```

```
redDot, = plt.plot([0], [np.sin(0)], 'ro')
```

```
def animate(i):
```

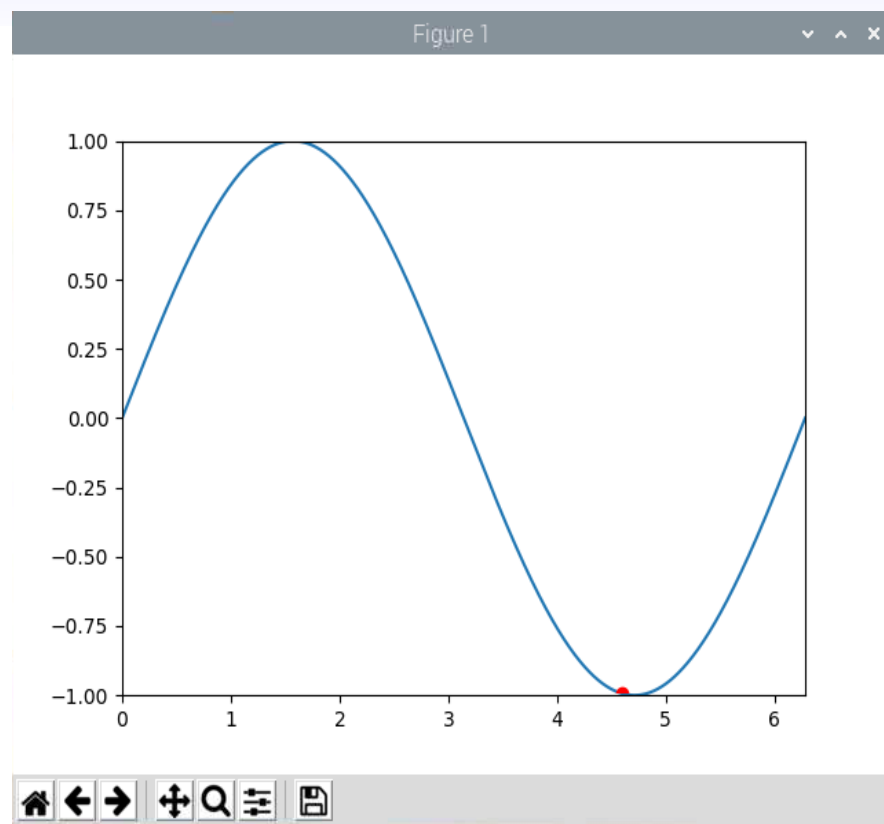
```
    redDot.set_data(i, np.sin(i))
```

```
    return redDot,
```

```
# create animation using the animate() function
```

```
myAnimation = animation.FuncAnimation(fig, animate, frames=np.arange(0.0, TWOPI, 0.1),
                                     interval=10, blit=True, repeat=True)
```

```
plt.show()
```



DHT11 습도 출력

```
# https://infinitt.tistory.com/40
from matplotlib import pyplot as plt
from matplotlib import animation
import numpy as np
import time
import Adafruit_DHT
sensor = Adafruit_DHT.DHT11
pin = 16

fig = plt.figure()
ax = plt.axes(xlim=(0, 50), ylim=(15, 45))
line, = ax.plot([], [], lw=1, c='blue', marker='d', ms=2)
max_points = 50
line, = ax.plot(np.arange(max_points),
                 np.ones(max_points, dtype=np.float)*np.nan, lw=1, c='blue', marker='d', ms=2)

def init():
    return line

h, t = Adafruit_DHT.read_retry(sensor, pin)

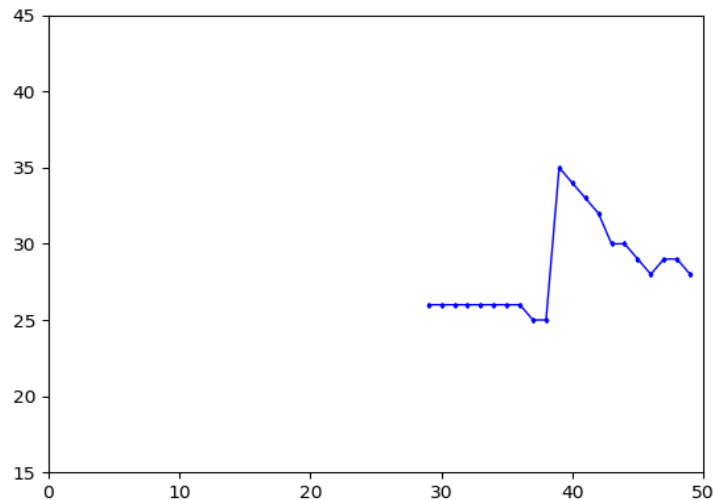
def get_y0 :
    h, t = Adafruit_DHT.read_retry(sensor, pin)
    return h
```

```
def animate(i):
```

```
    y = get_y0
```

```
    old_y = line.get_ydata()
    new_y = np.r_[old_y[1:], y]
    line.set_ydata(new_y)
    print(new_y)
    return line,
```

```
anim = animation.FuncAnimation(fig, animate, init_func=init, frames=200, interval=20, blit=False)
plt.show()
```



```
File Edit Tabs Help
nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan 26. 26. 26. 26. 26. 26. 26. 26. 25. 25. 35. 34.]
[nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan 26. 26. 26. 26. 26. 26. 26. 26. 25. 25. 35. 34. 33.]
[nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
26. 26. 26. 26. 26. 26. 26. 26. 25. 25. 35. 34. 33. 32.]
[nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
26. 26. 26. 26. 26. 26. 26. 26. 25. 25. 35. 34. 33. 32. 30.]
[nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
26. 26. 26. 26. 26. 25. 25. 35. 34. 33. 32. 30. 30. 29.]
[nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
26. 26. 26. 26. 25. 25. 35. 34. 33. 32. 30. 30. 29. 28. 29.]
[nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
26. 26. 26. 26. 25. 25. 35. 34. 33. 32. 30. 30. 29. 28. 29. 29.]
[nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
26. 26. 25. 25. 35. 34. 33. 32. 30. 30. 29. 28. 29. 29. 28.]
[nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
26. 25. 25. 35. 34. 33. 32. 30. 30. 29. 28. 29. 29. 28.]
```