

In [5]:

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

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

```
x = np.arange(0, 10, 0.1)
```

```
ims = []
```

```
for a in range(50):
```

```
    y = np.sin(x - a)
```

```
    im = plt.plot(x, y, "r")
```

```
    ims.append(im)
```

```
ani = animation.ArtistAnimation(fig, ims)
```

```
plt.show()
```

jupyter上で描かせるには必要

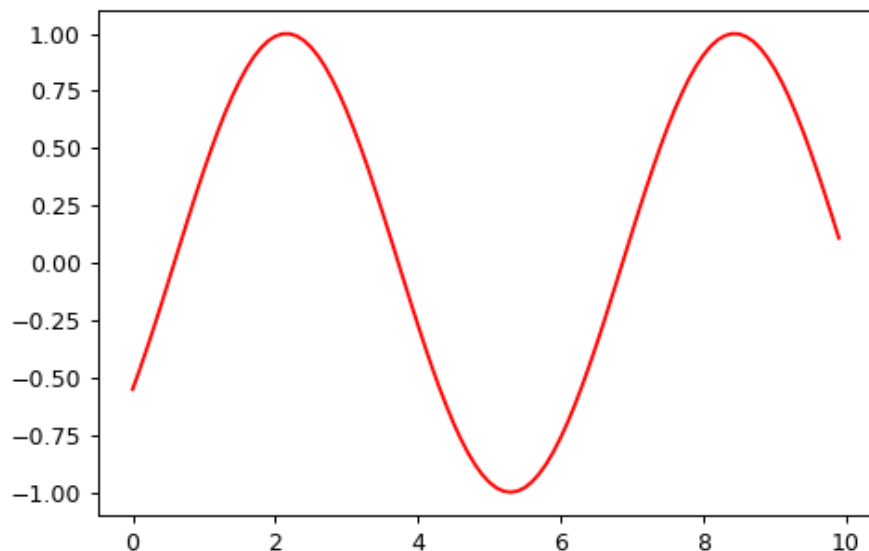
x軸0~10まで0.1刻みの数列

1ずつズレたxの値に対する
sinの値を50パターン算出してそ
の赤色のsin波図形を変数に代入

パターンを変数imsに格納

50個のパターンをパタパタの
アニメーションとして変数aniに
格納

表示



```
In [4]: %matplotlib notebook
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation

fig = plt.figure()
i = 0

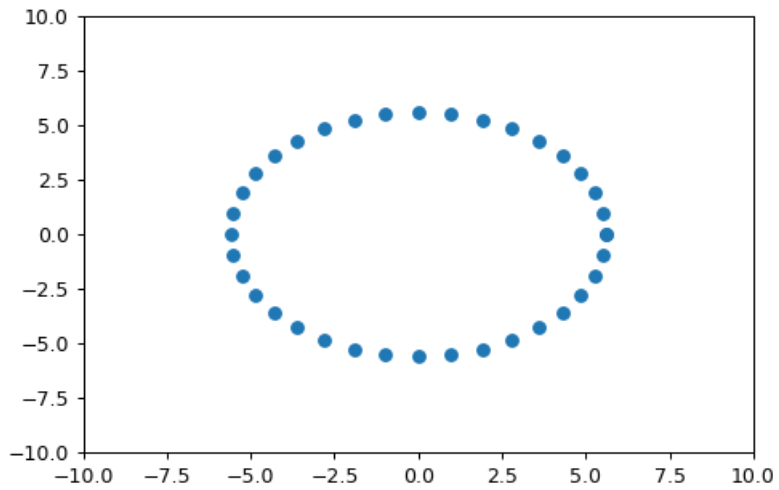
def plot(data):
    global i
    i = (i+1) % 100
    plt.cla()
    plt.xlim([-10,10])
    plt.ylim([-10,10])
    t = np.arange(0, 361, 10) * np.pi/180
    plt.scatter(np.cos(t)*i*0.1, np.sin(t)*i*0.1)

ani = animation.FuncAnimation(fig, plot, interval=30)
plt.show()
```

plot関数を定義

tの刻み(360°を10分割)

x=cos(t)、y=sin(t)で円を散布図として格納
plot関数で処理した散布図を30ミリ秒毎に描く



```
In [10]: %matplotlib notebook
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation

fig = plt.figure()
i = 0

def plot(data):
    global i
    i = (i+1) % 100
    plt.cla()
    plt.xlim([-10,10])
    plt.ylim([-10,10])
    t = np.arange(0, 361, 10) * np.pi/180
    x = np.cos(t)*i*0.1
    y = np.sin(t)*i*0.1
    u = np.cos(t)*i*0.1
    v = np.sin(t)*i*0.1
    plt.quiver(x,y,u,v,angles='xy',scale_units='xy',scale=1)

ani = animation.FuncAnimation(fig, plot, interval=10)
plt.show()
ani.save("test1.gif", writer="imagemagick")
```

(x,y)から(u,v)に向かうベクトルを描く

(x,y)と(u,v)に円の座標データを代入

アニメーションをgifファイルで保存

