

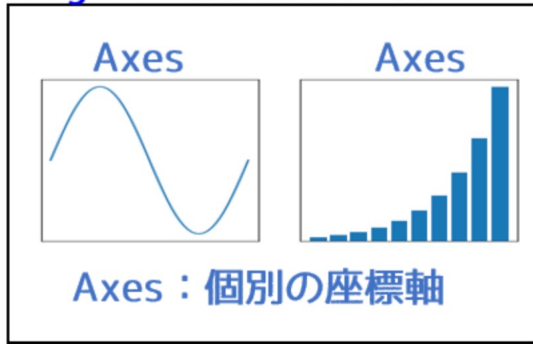
関数プロット

test_1.py ▾



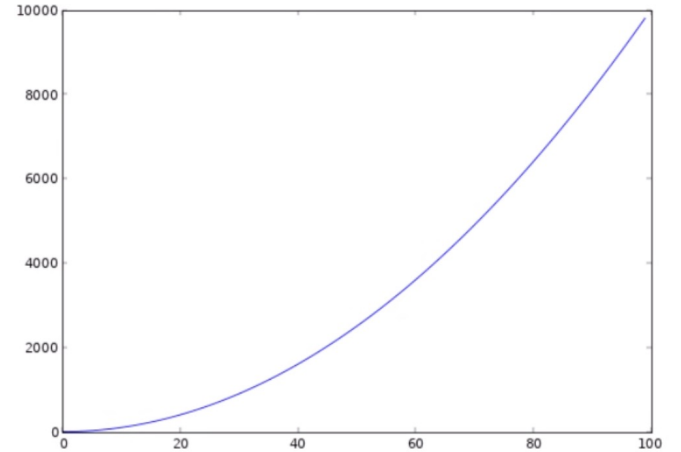
```
1 import numpy as np
2 import matplotlib.pyplot as
  plt
3 x=np.arange(100)
4 y=x**2
5 Figure,ax=plt.subplots()
6 ax.plot(x,y)
7 plt.show()
```

Figure : 描画領域全体



> Console ▾

Clear

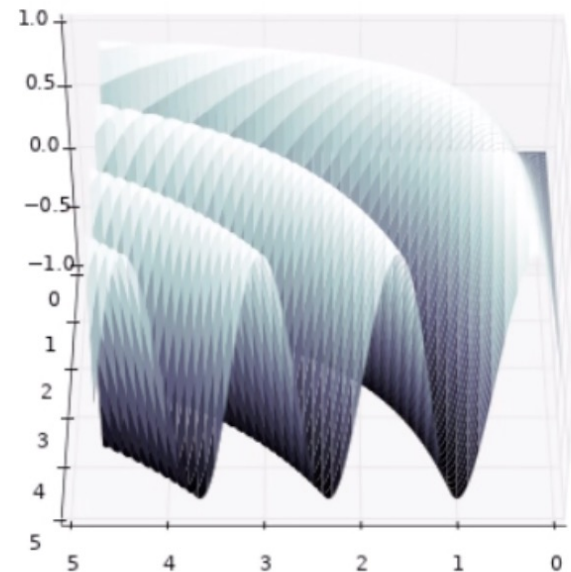


FigureとAxesを
一緒に済ませる

Figure や Axes を一つずつ作成していく場合は、

- `fig = plt.figure()` ⇒ Figure を生成
- `fig.add_subplot()` ⇒ Axes を1個ずつ追加 or `fig.subplots()` ⇒ Axes を複数追加
- `fig, ax = plt.subplots()` 引数を省力 : 1つのサブプロットを生成
- `fig, axes = plt.subplots(nrows, ncols)` : `nrows` × `ncols` 個のサブプロットを生成

3Dプロット



test3_1.py

test3_1.py

```
1 import numpy as np
2 import matplotlib.pyplot as
  plt
3 from mpl_toolkits.mplot3d
  import Axes3D
4
5 ax=plt.figure().gca(projection
  = '3d')
6 xarr=np.arange(0,np.pi*1.5,0.0
  1)
7 yarr=np.arange(0,np.pi*1.5,0.0
  1)
8 (x,y)=np.meshgrid(xarr,yarr)
9 z=np.sin(x*y)
10
11 sutf=ax.plot_surface(x,y,z,cma
  p='bone',linewidth=0,antialias
  ed=True)
12 ax.view_init(45,90)
13 plt.show()
```

←3Dプロットに必要

←figureのaxにprojection='3d'を適用して3Dプロットが生成

• numpy.arange(start, stop, step)

◦ start ≤ n < stop で間隔は step

0 ≤ n ≤ 1.5π 0.01間隔

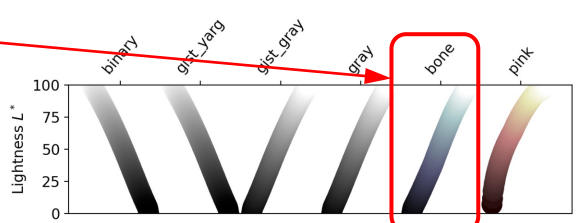
←2次元配列生成

2次元配列である必要

ax.plot_surface(x, y, z, cmap, antialiased=True)

color map 色指定

なめらかにする場合True



ヒストグラム

```
test_1.py × test2.py
1 import numpy as np
2 import matplotlib.pyplot as
. plt
3 x=np.random.randn(1000)*100
4 plt.hist(x,20)
5 plt.show()
```

階級数

平均値0、標準偏差1の
1000個の乱数

> Console ▾



Clear

