In [8]:

```
import numpy as np
a=np.arange(5)
b=np.arange(4,-1,-1)
```

In [9]:

```
a
```

Out[9]:

```
array([0, 1, 2, 3, 4])
```

In [10]:

```
b
```

Out[10]:

```
array([4, 3, 2, 1, 0])
```

In [11]:

```
np.any(a==b)
```

Out[11]:

```
True
```

In [12]:

```
np.any(a==b)and np.any(a>b)
```

Out[12]:

```
True
```

In [13]:

```
np.all(a==b)
```

Out[13]:

```
False
```

In [14]:

```
(a==b)|(a>b)
```

Out[14]:

```
array([False, False,  True,  True,  True], dtype=bool)
```

In [15]:

```
~np.arange(5)
```

Out[15]:

```
array([-1, -2, -3, -4, -5])
```

In [17]:

```
~np.arange(5,dtype=np.uint8)
```

Out[17]:

```
array([255, 254, 253, 252, 251], dtype=uint8)
```

In [18]:

```
np.arange(5)
```

Out[18]:

```
array([0, 1, 2, 3, 4])
```

In [19]:

```
~np.arange(2,9,1)
```

Out[19]:

```
array([-3, -4, -5, -6, -7, -8, -9])
```

自定义函数

In [36]:

```
def triangle_wave(x,c,c0,hc):
    x=x-int(x)
    if x>=c:y=0.0
    elif x<c0:y=x/c0*hc
    else:y=(c-x/(c-c0))*hc
    return y
```

In [40]:

```
x=np.linspace(0,2,1000)
y1=np.array([triangle_wave(t,0.6,0.4,1.0)for t in x])
```

In [42]:

```
triangle_ufunc1=np.frompyfunc(triangle_wave,4,1)
y2=triangle_ufunc1(x,0.6,0.4,1.0)
```

In [43]:

```
y2.dtype
```

Out[43]:

```
dtype('O')
```

In [44]:

```
np.all(y1==y2)
```

Out[44]:

```
True
```

广播

In [47]:

```
a=np.arange(0,60,10).reshape(6,1)
```

In [48]:

```
a
```

Out[48]:

```
array([[ 0],
       [10],
       [20],
       [30],
       [40],
       [50]])
```

In [49]:

```
a.shape
```

Out[49]:

```
(6, 1)
```

In [50]:

```
b=np.arange(1,6)
```

In [51]:

```
b
```

Out[51]:

```
array([1, 2, 3, 4, 5])
```

In [52]:

```
b.shape
```

Out[52]:

```
(5,)
```

In [54]:

```
c=a+b
c
```

Out[54]:

```
array([[ 1,  2,  3,  4,  5],
       [11, 12, 13, 14, 15],
       [21, 22, 23, 24, 25],
       [31, 32, 33, 34, 35],
       [41, 42, 43, 44, 45],
       [51, 52, 53, 54, 55]])
```

In [56]:

```
c.shape
```

Out[56]:

```
(6, 5)
```

In [59]:

```
b=b.repeat(6,axis=0)
b
```

Out[59]:

```
array([1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 4, 4, 4,
4, 4,
       4, 5, 5, 5, 5, 5, 5])
```

In [60]:

```
b.shape
```

Out[60]:

```
(30,)
```

In [61]:

```
a=a.repeat(5,axis=1)
a
```

Out[61]:

```
array([[ 0,  0,  0,  0,  0],
       [10, 10, 10, 10, 10],
       [20, 20, 20, 20, 20],
       [30, 30, 30, 30, 30],
       [40, 40, 40, 40, 40],
       [50, 50, 50, 50, 50]])
```

In [62]:

```
a.shape
```

Out[62]:

```
(6, 5)
```

In [64]:

```
x,y=np.ogrid[:5,:5]
x
```

Out[64]:

```
array([[0],
       [1],
       [2],
       [3],
       [4]])
```

In [65]:

```
y
```

Out[65]:

```
array([[0, 1, 2, 3, 4]])
```

In [66]:

```
x,y=np.mgrid[:5,:5]
```

In [67]:

```
x
```

Out[67]:

```
array([[0, 0, 0, 0, 0],
       [1, 1, 1, 1, 1],
       [2, 2, 2, 2, 2],
       [3, 3, 3, 3, 3],
       [4, 4, 4, 4, 4]])
```

In [68]:

```
y
```

Out[68]:

```
array([[0, 1, 2, 3, 4],
       [0, 1, 2, 3, 4],
       [0, 1, 2, 3, 4],
       [0, 1, 2, 3, 4],
       [0, 1, 2, 3, 4]])
```

In [69]:

```
x,y=np.ogrid[:1:4j,:1:3j]
x
```

Out[69]:

```
array([[ 0.        ],
       [ 0.33333333],
       [ 0.66666667],
       [ 1.        ]])
```

In [70]:

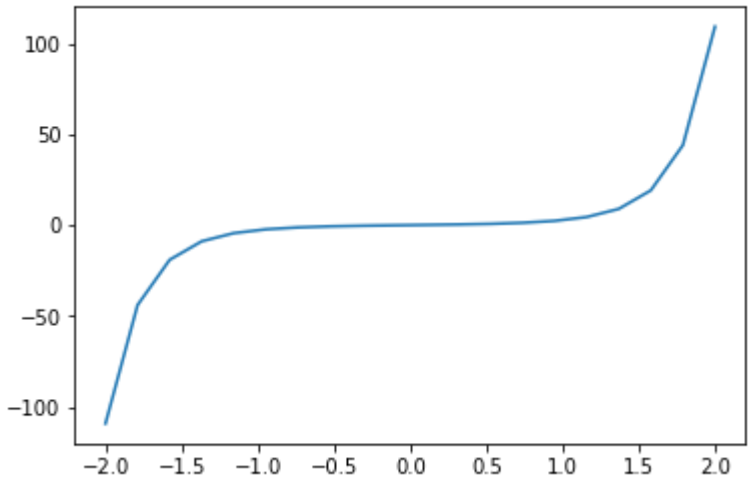
```
y
```

Out[70]:

```
array([[ 0. ,  0.5,  1. ]])
```

In [80]:

```
import matplotlib.pyplot as plt
x=np.ogrid[-2:2:20j]
y=x*np.exp(x**2)
plt.plot(x,y)
plt.show()
```



In [81]:

```
a=np.arange(4)
a[None,:]
```

Out[81]:

```
array([[0, 1, 2, 3]])
```

In [82]:

```
a[:,None]
```

Out[82]:

```
array([[0],
       [1],
       [2],
       [3]])
```

In [83]:

```
x=np.array([0,1,4,10])
y=np.array([2,3,4])
x[None,:]+y[:,None]
```

Out[83]:

```
array([[ 2,  3,  6, 12],
       [ 3,  4,  7, 13],
       [ 4,  5,  8, 14]])
```

In [84]:

```
gy,gx=np.ix_(y,x)#ix_(将两个一维数组转化为可广播的二维数组)
```

In [85]:

```
gx
```

Out[85]:

```
array([[ 0,  1,  4, 10]])
```

In [86]:

```
gy
```

Out[86]:

```
array([[2],
       [3],
       [4]])
```

In [87]:

```
gy+gx
```

Out[87]:

```
array([[ 2,  3,  6, 12],
       [ 3,  4,  7, 13],
       [ 4,  5,  8, 14]])
```

In [ ]:

```

```