Examples (/examples) › Python Plots (/examples/python-plots) › Logarithmic Axes

This example shows how to create plots with logarithmic scales on either one of the axes or both of them. It is even possible to use a different base on each axes.

First, the required modules are imported. The array-manipulation module **numpy** and the matplotlib submodule **pyplot**, to plot 2d graphics. The corresponding aliases `np` and `plt` for these two modules are widely used python conventions. The numbers in array `t` are the points to plot on the x-axis.

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

t = np.arange(0.01, 20.0, 0.01)
```

Logarithmic Scale On The y Axis

The command `semilogy` will enable logarithmic scale on the y-axis, the additional option `basey` can be used to manually set the logarithm base

```
plt.figure()
plt.semilogy(t, np.exp(-t/5.0),
             color='purple',
             linewidth = 2)
plt.title('Semilog y-axis')
plt.grid(True)
```

Logarithmic Scale On The x Axis

The command `semilogx` will enable logarithmic scale on the y-axis, the additional option `basex` manually sets the logarithm base to use.

```
plt.figure()
plt.semilogx(t, np.sin(2*np.pi*t),
             basex=10,
             color='darkred',
             linewidth = 0.5)
plt.title('Semilog x-axis')
plt.grid(True)
```

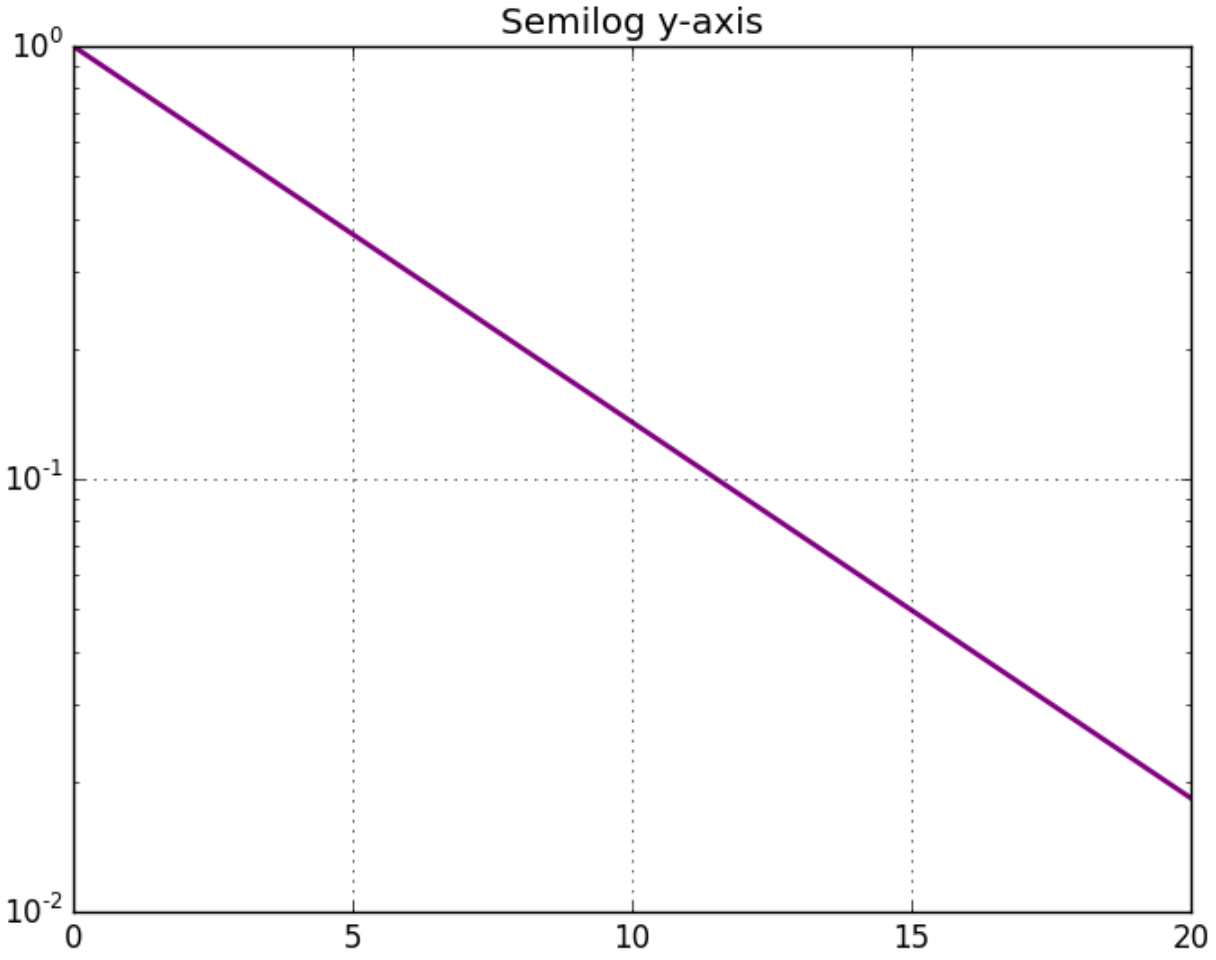
Logarithmic Scale On Both Axes

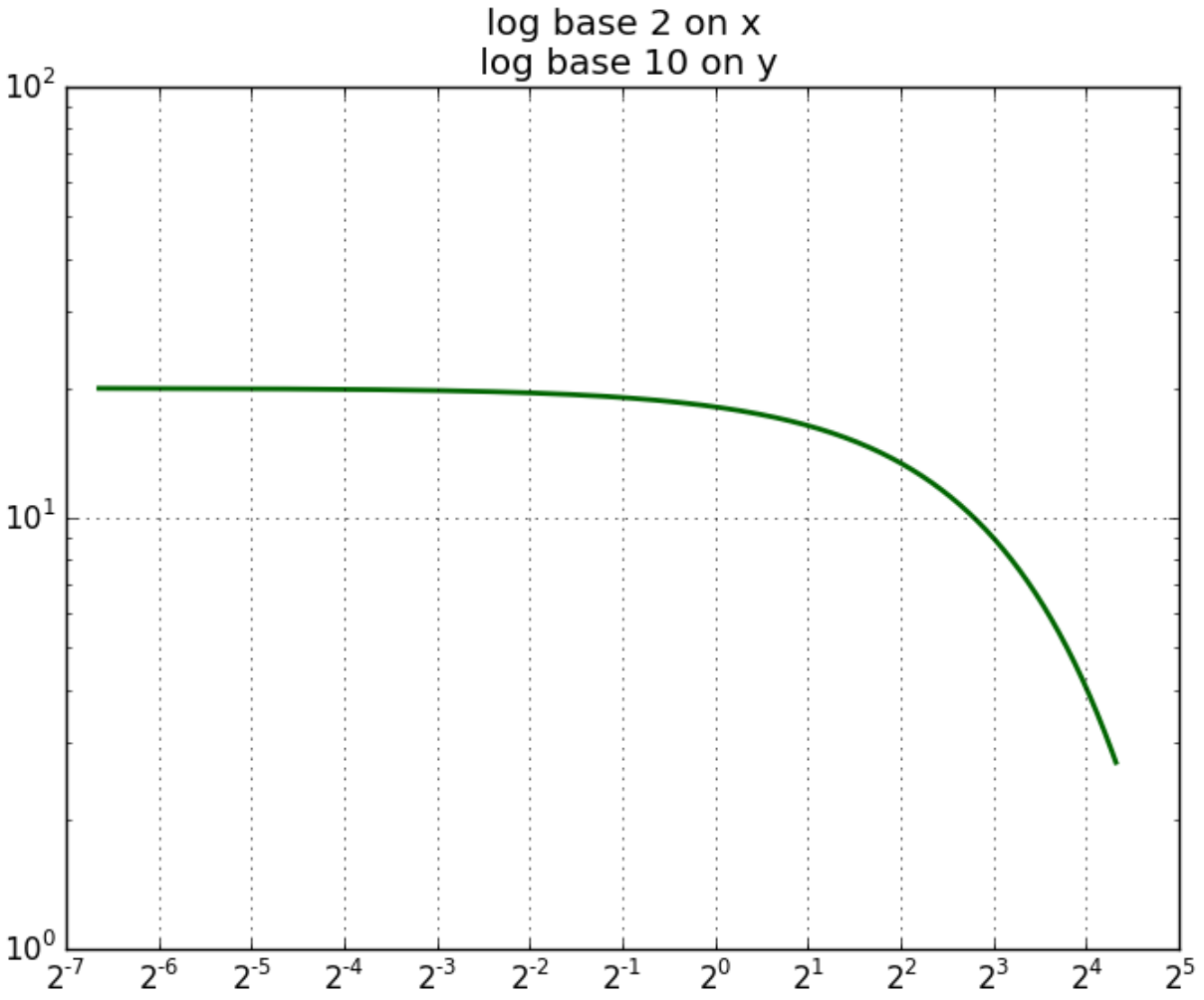
The command `loglog()` will use logarithmic scales on both axes, the base for each axes can be set with the `basex` and `basey` parameters.

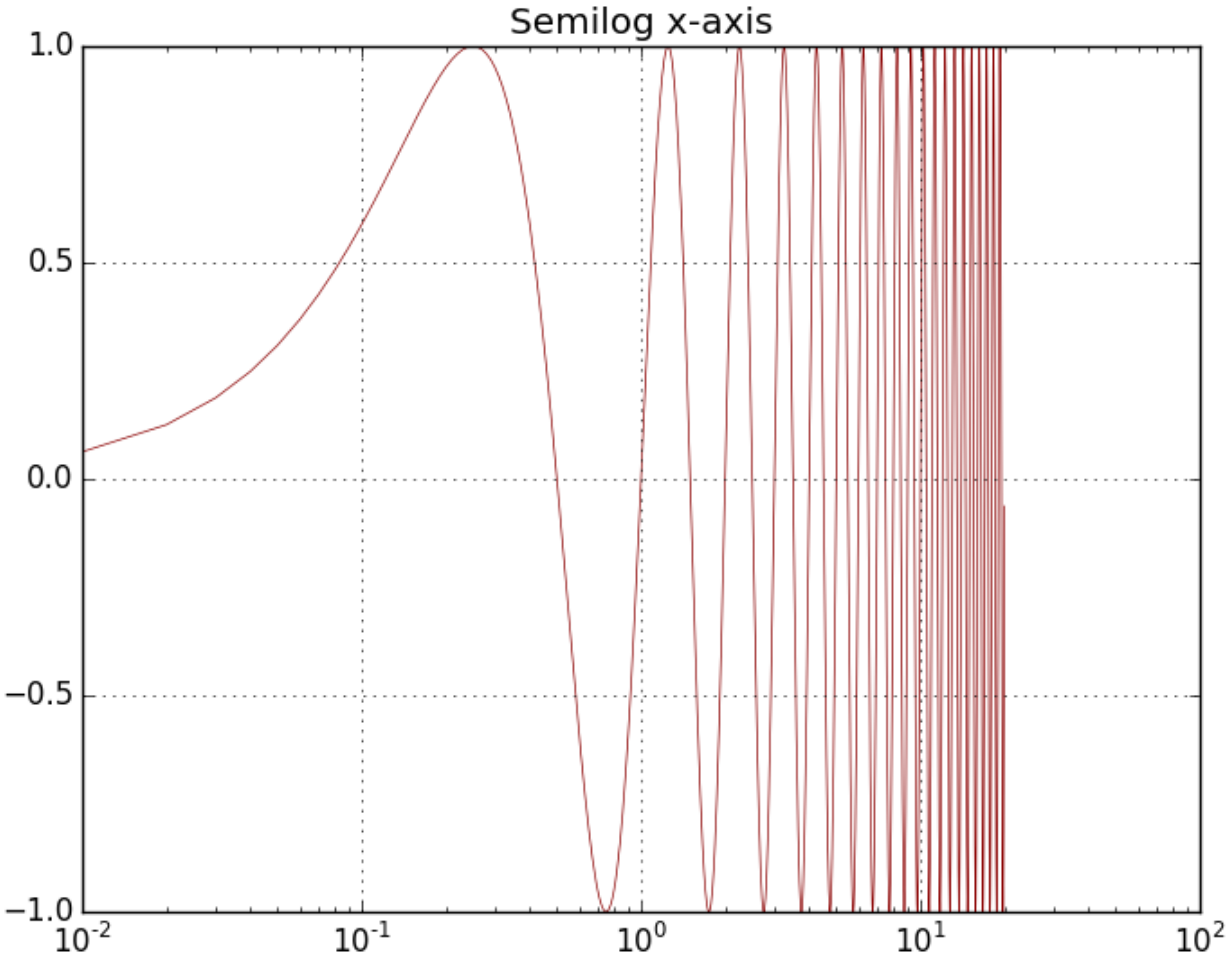
```
plt.figure()
plt.loglog(t, 20*np.exp(-t/10.0),
           basex=2,
           color='darkgreen',
           linewidth = 2)
plt.grid(True)
plt.title('log base 2 on x \n log base 10 on y')

plt.show()
```

Output


Semilog y-axis
10^0
10^{-1}
10^{-2}
0
5
10
15
20


log base 2 on x
log base 10 on y
10^2
10^1
10^0
2^{-7} 2^{-6} 2^{-5} 2^{-4} 2^{-3} 2^{-2} 2^{-1} 2^0 2^1 2^2 2^3 2^4 2^5



Comment