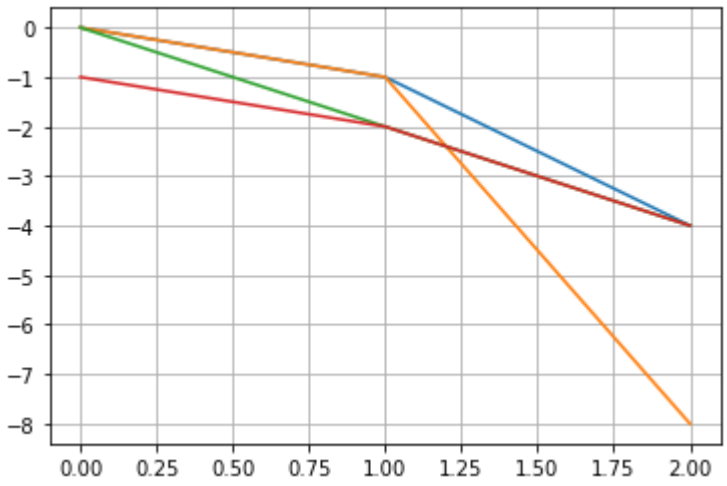
In [1]:

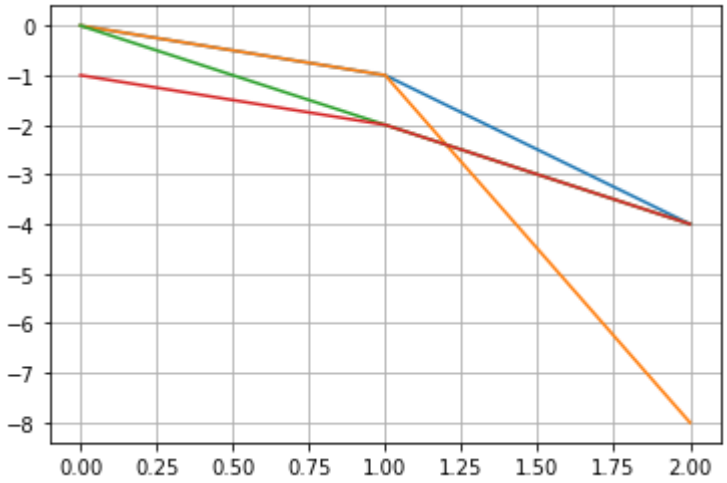
```
# Grid, Axes, and Labels

# Here, you will learn how to enable a grid in the visualizations. This can be done with
# statement plt.grid(True). You will also learn how to manipulate the limits of axes. Bu
# before that, you will quickly learn how to save a visualization as an image on the har
# disk. Look at the following code:

import numpy as np
import matplotlib.pyplot as plt
x = np.arange(3)
plt.plot(x, -x**2, x, -x**3, x, -2*x, x, -2**x)
plt.grid(True)
plt.savefig('test.png')
plt.show()

# The statement plt.savefig('test.png') saves the image in the current directory of
# the Jupyter Notebook file.
```



In [2]:

```
# You can see that the limits of the axes are set by default as follows:

import numpy as np
import matplotlib.pyplot as plt
x = np.arange(3)
plt.plot(x, -x**2, x, -x**3, x, -2*x, x, -2**x)
plt.grid(True)
print(plt.axis())
plt.show()
```

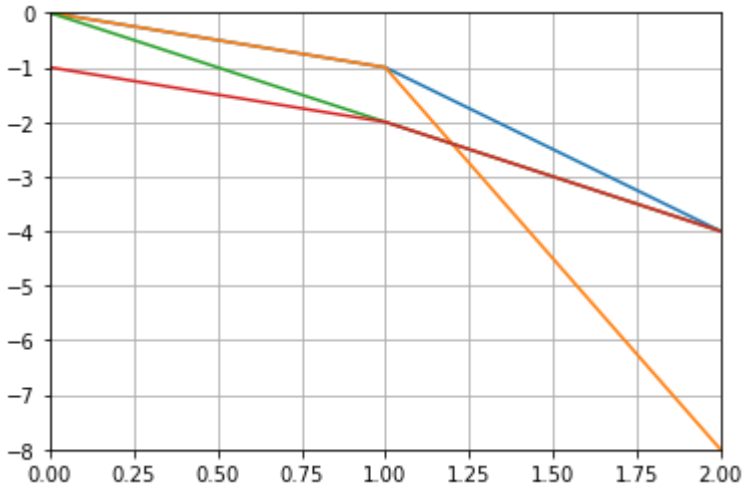
```
(-0.1, 2.1, -8.4, 0.4)
```



In [3]:

```
# You can also customize the values of the axes as follows:

import numpy as np
import matplotlib.pyplot as plt
x = np.arange(3)
plt.plot(x, -x**2, x, -x**3, x, -2*x, x, -2**x)
plt.grid(True)
plt.axis([0, 2, -8, 0])
print(plt.axis())
plt.show()

# The statement plt.axis([0, 2, -8, 0]) sets the values of the axes. The first pair,
# (0, 2), refers to the limits for the x-axis, and the second pair, (-8, 0),
# refers to the limits for the y-axis.
```

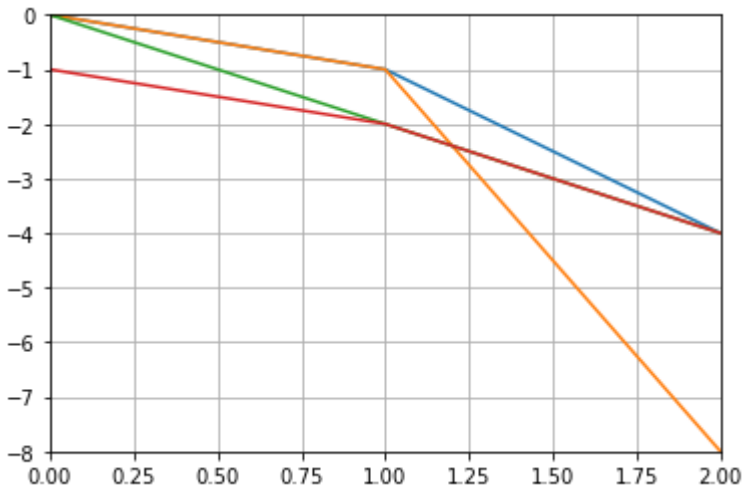
```
(0.0, 2.0, -8.0, 0.0)
```



In [4]:

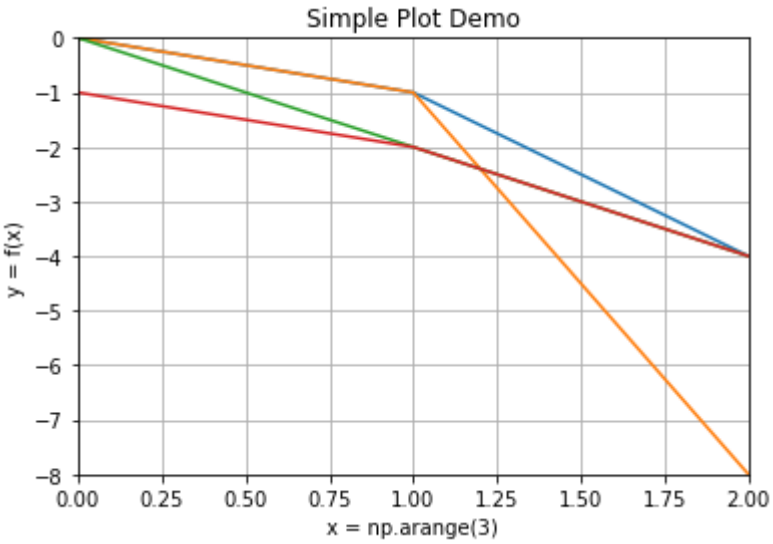
```
# You can write the previous code with different syntax using the functions
# xlim() and ylim() as follows:

import numpy as np
import matplotlib.pyplot as plt
x = np.arange(3)
plt.plot(x, -x**2, x, -x**3, x, -2*x, x, -2**x)
plt.grid(True)
plt.xlim([0, 2])
plt.ylim([-8, 0])
print(plt.axis())
plt.show()
```

```
(0.0, 2.0, -8.0, 0.0)
```



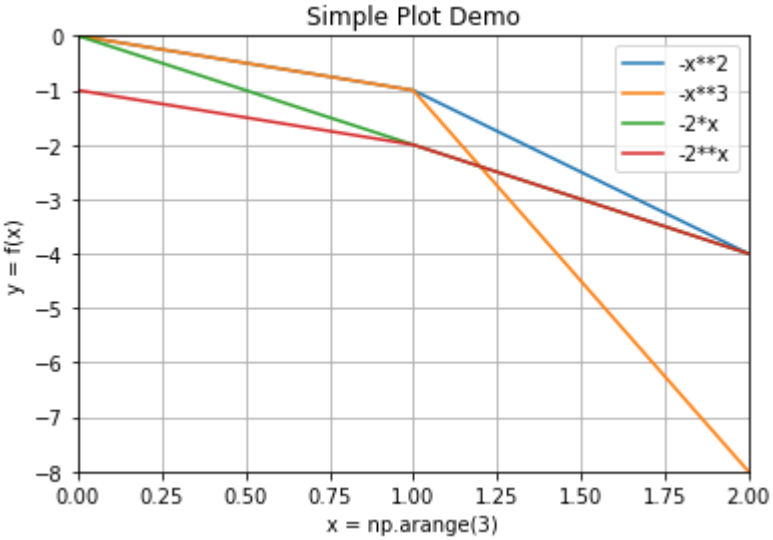
In [5]:

```
# You can add the title and the labels for the axes as follows:

import numpy as np
import matplotlib.pyplot as plt
x = np.arange(3)
plt.plot(x, -x**2, x, -x**3, x, -2*x, x, -2**x)
plt.grid(True)
plt.xlabel('x = np.arange(3)')
plt.xlim([0, 2])
plt.ylabel('y = f(x)')
plt.ylim([-8, 0])
plt.title('Simple Plot Demo')
plt.show()
```



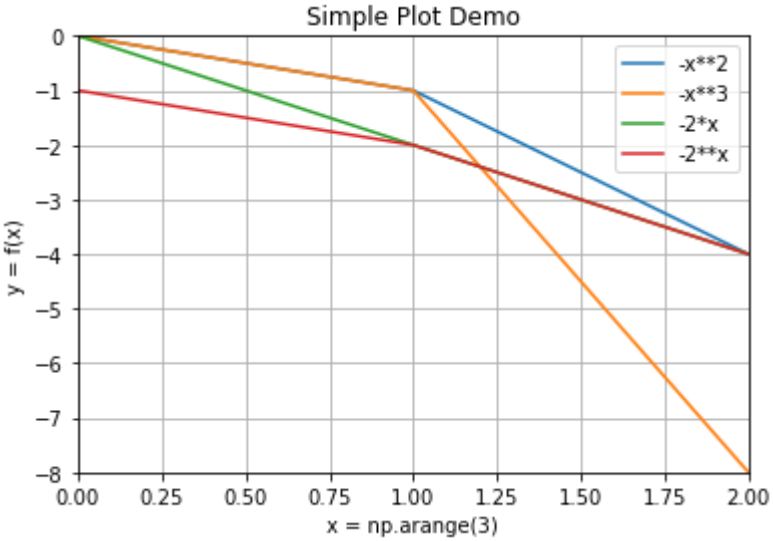
In [7]:

```
# You can pass an argument for the parameter label in the plot() function and then
# call the function legend() to create a legend as follows:

import numpy as np
import matplotlib.pyplot as plt
x = np.arange(3)
plt.plot(x, -x**2, label='-x**2')
plt.plot(x, -x**3, label='-x**3')
plt.plot(x, -2*x, label='-2*x')
plt.plot(x, -2**x, label='-2**x')
plt.legend()
plt.grid(True)
plt.xlabel('x = np.arange(3)')
plt.xlim([0, 2])
plt.ylabel('y = f(x)')
plt.ylim([-8, 0])
plt.title('Simple Plot Demo')
plt.show()
```



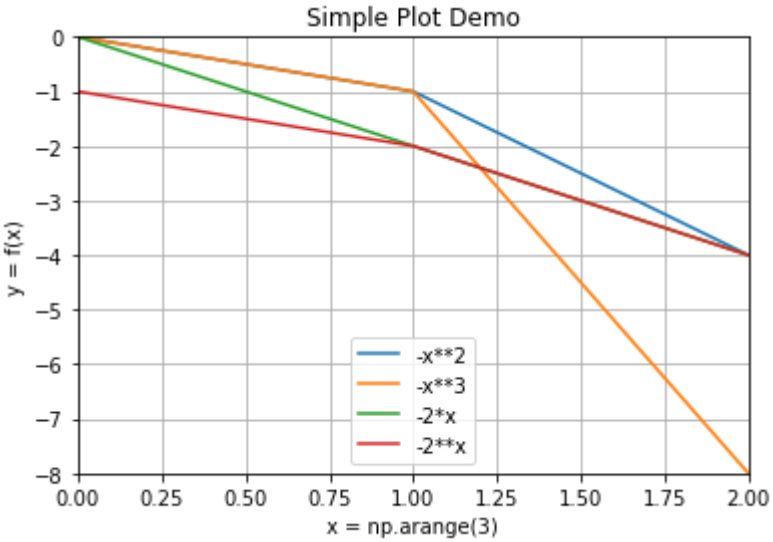
In [8]:

```
# Instead of passing the legend string as an argument to the function plot(), you can
# pass the list of strings as an argument to the function legend() as follows:

import numpy as np
import matplotlib.pyplot as plt
x = np.arange(3)
plt.plot(x, -x**2, x, -x**3, x, -2*x, x, -2**x)
plt.legend(['-x**2', '-x**3', '-2*x', '-2**x'])
plt.grid(True)
plt.xlabel('x = np.arange(3)')
plt.xlim([0, 2])
plt.ylabel('y = f(x)')
plt.ylim([-8, 0])
plt.title('Simple Plot Demo')
plt.show()
```



In [10]:

```
# You can also change the location of the legend box by making the following changes
# to plt.legend() from the previous code:

import numpy as np
import matplotlib.pyplot as plt
x = np.arange(3)
plt.plot(x, -x**2, x, -x**3, x, -2*x, x, -2**x)
plt.legend(['-x**2', '-x**3', '-2*x', '-2**x'],
loc='lower center')
plt.grid(True)
plt.xlabel('x = np.arange(3)')
plt.xlim([0, 2])
plt.ylabel('y = f(x)')
plt.ylim([-8, 0])
plt.title('Simple Plot Demo')
plt.show()
```



In [11]:

```
# Finally, let's save the visualization to disk with the following code:

import numpy as np
import matplotlib.pyplot as plt
x = np.arange(3)
plt.plot(x, -x**2, x, -x**3, x, -2*x, x, -2**x)
plt.legend(['-x**2', '-x**3', '-2*x', '-2**x'],
loc='lower center')
plt.grid(True)
plt.xlabel('x = np.arange(3)')
plt.xlim([0, 2])
plt.ylabel('y = f(x)')
plt.ylim([-8, 0])
plt.title('Simple Plot Demo')
plt.savefig('test.png')
plt.show()
```