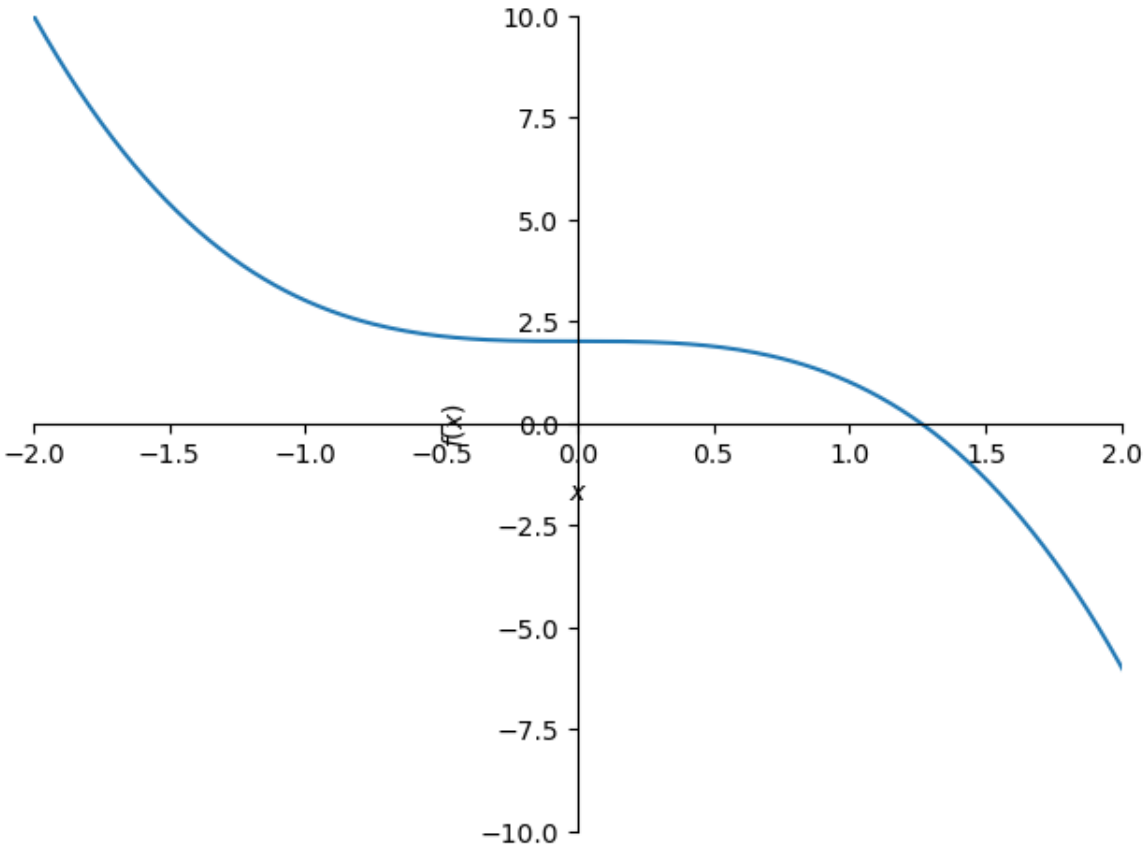
In [4]:
```
from __future__ import print_function
%matplotlib inline
#import ganymede
#ganymede.configure('uav.beaver.works')
import matplotlib.pyplot as plt
import numpy as np
import sympy as sym
sym.init_printing(use_latex = "mathjax")
```

Enter your name below and run the cell:Joy Deng

Individual cells can be run with `Ctrl + Enter`

In [6]:
```
#ganymede.name('YOUR NAME HERE')
#def check(p):
#    ganymede.update(p,True)
#check(0)
```

In [7]:
```
x = sym.symbols('x')
expr = -x ** 3 + 2

sym.plot(expr, xlim=(-2, 2), ylim=(-10, 10));
```



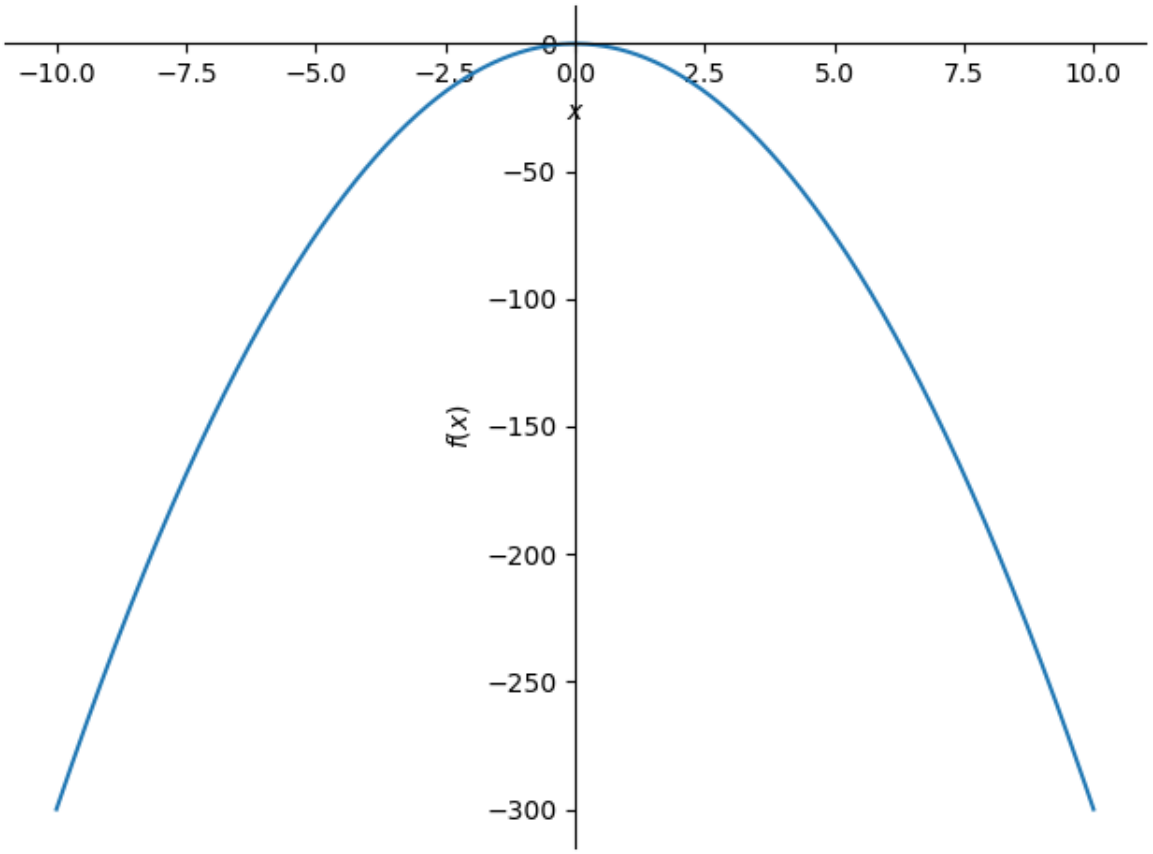
In [8]:
```
sym.Derivative(expr)
```

Out[8]: $\frac{d}{dx}\left(2 - x^{3}\right)$

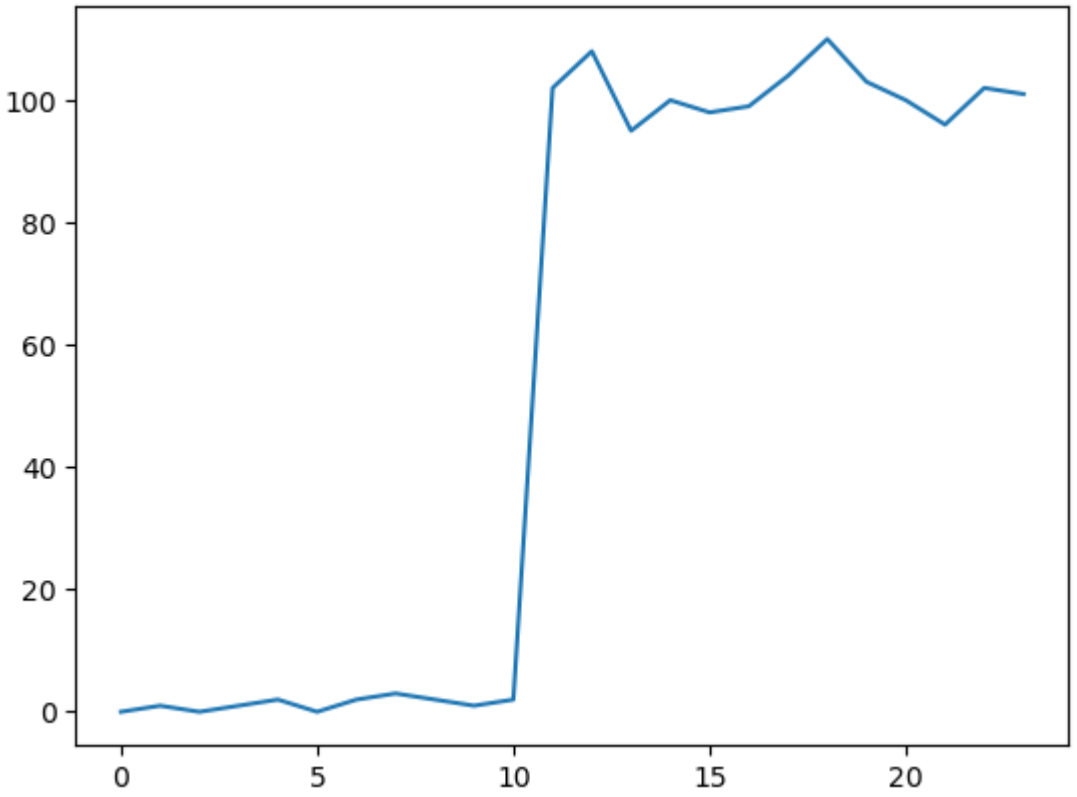
In [9]:
```
sym.Derivative(expr).doit()
```

Out[9]: $-3x^2$

In [10]:
```
sym.plot(sym.diff(expr));
```



In [11]:
```
ys = np.array([0, 1, 0, 1, 2, 0, 2, 3, 2, 1, 2, 102, 108,  95, 100,  98,

fig,ax = plt.subplots()
ax.plot([i for i in range(len(ys))], ys);
```

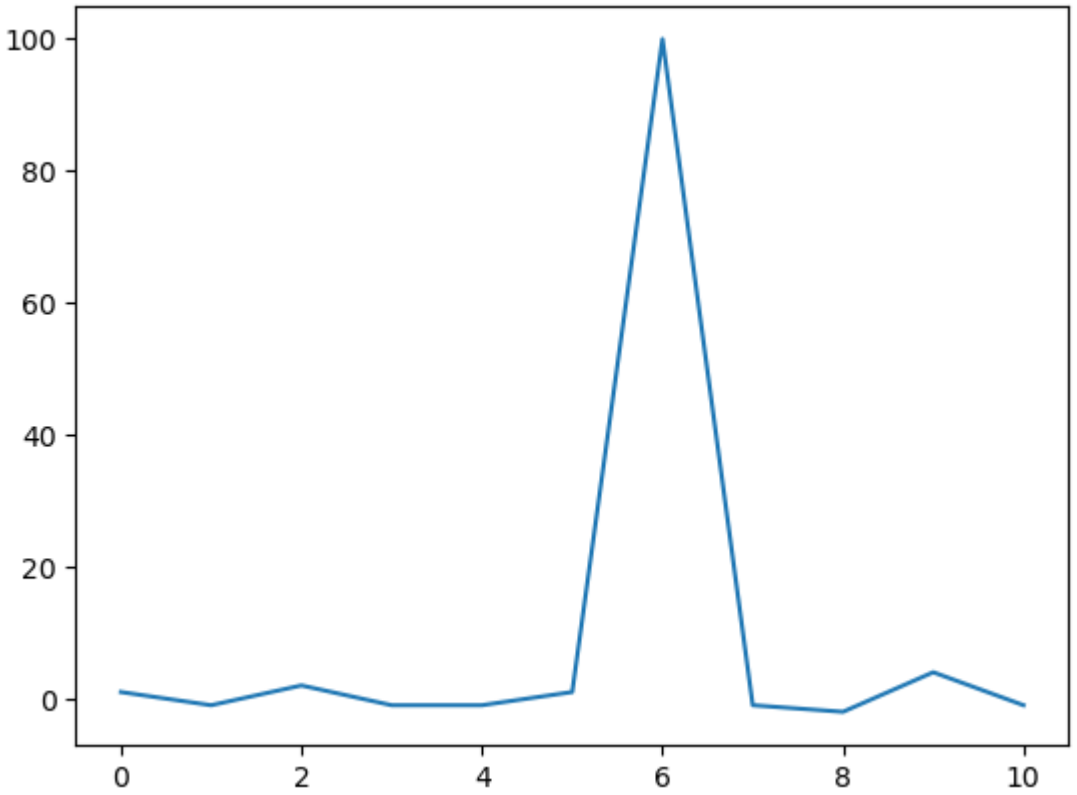


In [12]:
```
def make_windows(sequence, windowsize):
    positions = len(sequence) - windowsize + 1
    windows = []
    for i in range(positions):
        windows.append(sequence[i:i+windowsize])
    return windows

def print_padded_seq(seq):
    print("[", ",".join(["{:4d}".format(i) for i in seq]), ']')

def print_sliding_windows(seq, windowsize=3):
    windows = make_windows(seq, windowsize)
    for window in windows:
        print(",".join(["{:4d}".format(i) for i in window]))
```

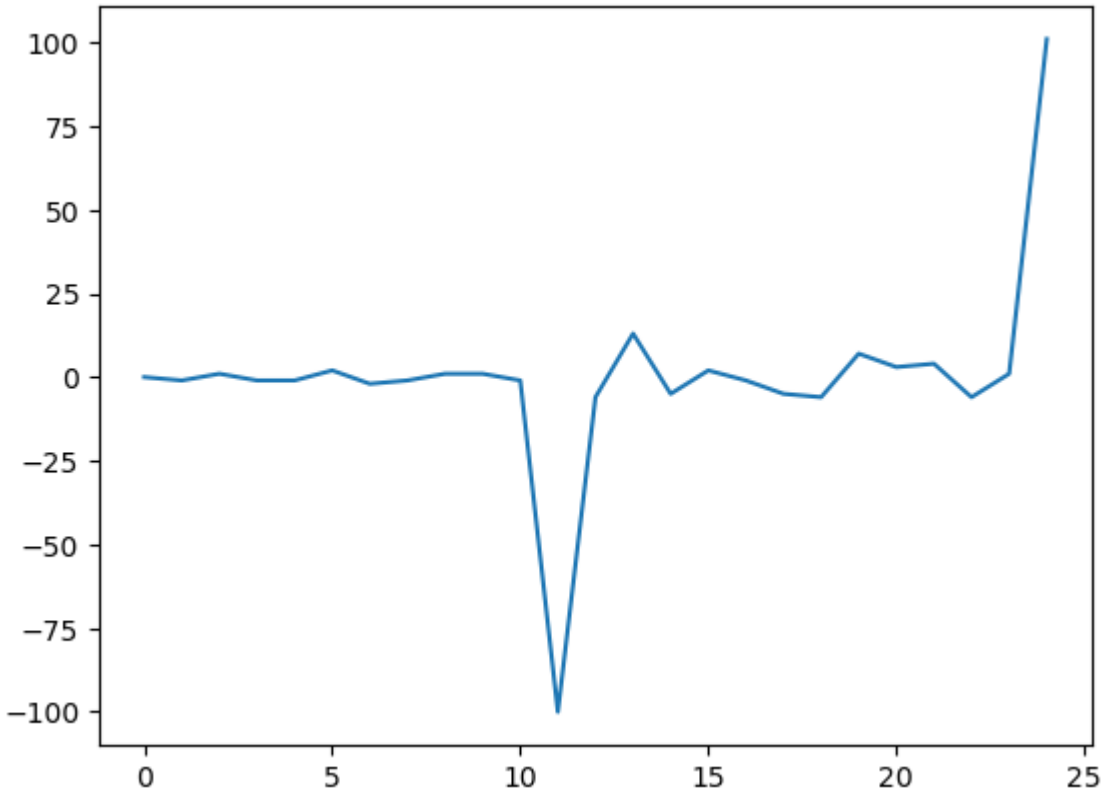
In [18]:
```
series = [0, 1, 0, 2, 1, 0, 1, 101, 100, 98, 102, 101]
windowsize = 2

print_padded_seq(series)

print_sliding_windows(series, windowsize=windowsize)
```

```
[    0,   1,   0,   2,   1,   0,   1, 101, 100,  98, 102, 101 ]
   0,   1
   1,   0
   0,   2
   2,   1
   1,   0
   0,   1
   1, 101
 101, 100
 100,  98
  98, 102
 102, 101
```

In [17]:
```
convolutions = []
kernel = np.array([-1,1])
for w in make_windows(series, windowsize=2):
    w = np.array(w)
    convolved = np.dot(w,kernel)
    convolutions.append(convolved)

plt.plot(convolutions);
```



In [16]:
```
convolved = np.convolve([-1, 1], ys)
ys = np.array([0, 1, 0, 1, 2, 0, 2, 3, 2, 1, 2, 102, 108,  95, 100,  98,

fig,ax = plt.subplots()
ax.plot([i for i, _ in enumerate(convolved)], convolved);
```



Question: Why does the graph move up at the end? Because convolve funtion describes the overlapping in a numpy array, and it assumes that theres a zero at both ends, so it essentially graphs, 0, -1... and lastly 101