In [8]:

```
from __future__ import print_function, division

%matplotlib inline

import thinkdsp
import thinkplot
import numpy as np
import math

from ipywidgets import interact, interactive, fixed
import ipywidgets as widgets
PI2 = 2 * math.pi
```

In [9]:

```
#Exer 3.3
```

In [10]:

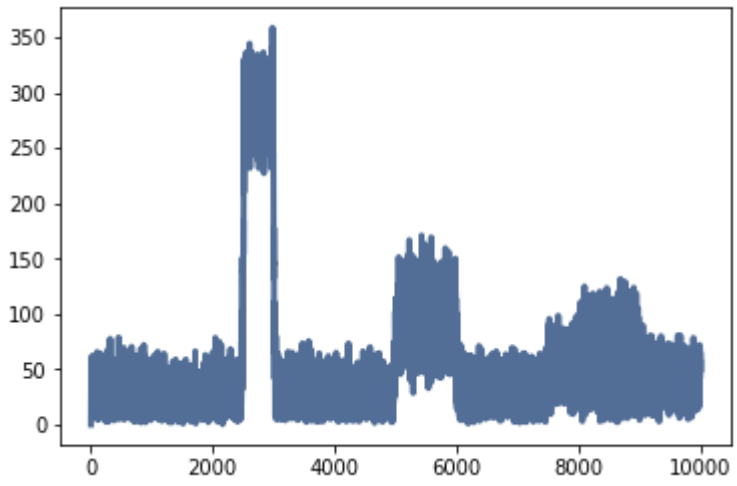
```
class SawtoothChirp(thinkdsp.Chirp):


    def _evaluate(self, ts, freqs):#private method

        dts = np.diff(ts)#인접한 ts의 원소의 차를 구한다.
        dphis=PI2*freqs*dts#phase 생성
        phases=np.cumsum(dphis)#phase의 total
        phases=np.insert(phases,0,0)# phase의 0번째 원소에 0추가
        cycles = phases / PI2
        frac, _ = np.modf(cycles)
        ys = thinkdsp.normalize(thinkdsp.unbias(frac), self.amp)
        return ys# thinkdsp.py에 없는 SawtoothChirp클래스 만들기
```

In [11]:

```
signal=SawtoothChirp(start=2500, end=3000)
wave=signal.make_wave(duration=1, framerate=20000)
spectrum=wave.make_spectrum()#spectrum 만들기
spectrum.plot()#스펙트럼 그리기
```



In [ ]: