

디지털 신호 처리 3주차 과제

201710758 휴먼지능정보공학과 김진성

```
In [74]: from thinkdsp import play_wave, read_wave
from IPython.display import Audio
import thinkplot
```

다운로드 한 Sound sample

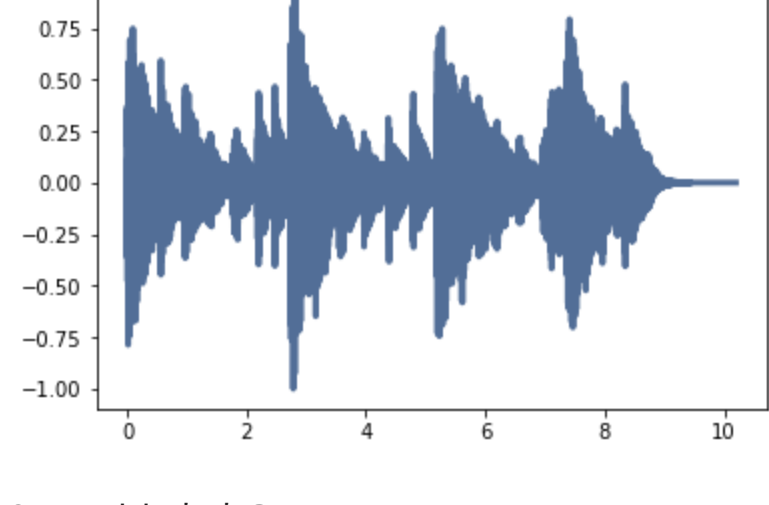
```
In [75]: wave = read_wave('piano_sound.wav')
wave.make_audio()
```

Out[75]:

▶ 0:00 / 0:10

🔊 ⋮

```
In [76]: wave.plot()
```



0.5초 분량의 Segment

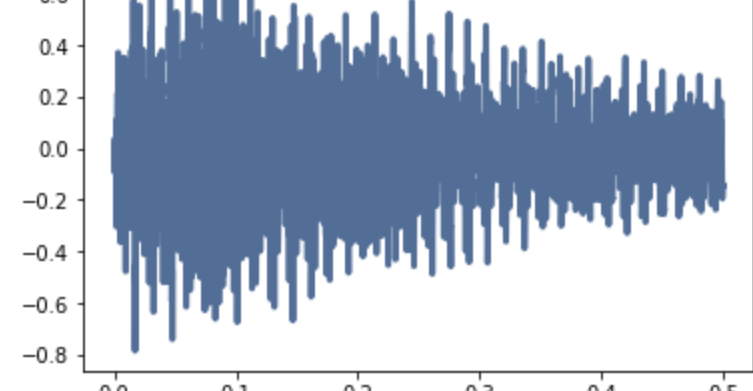
```
In [77]: start = 0
duration = 0.5
segment = wave.segment(start, duration)
segment.make_audio()
```

Out[77]:

▶ 0:00 / 0:00

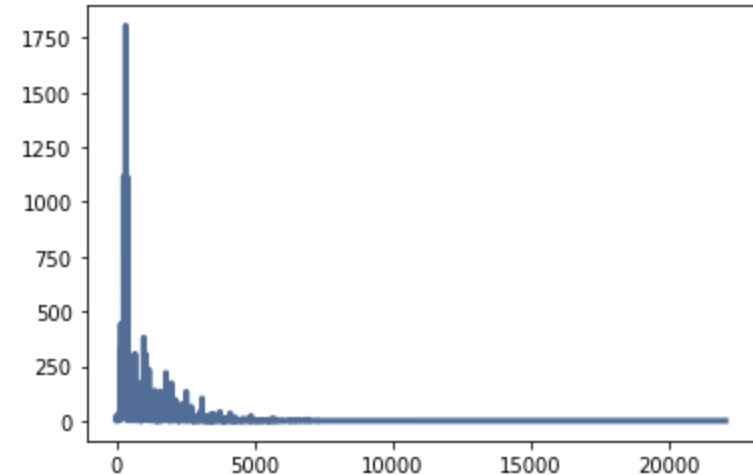
🔊 ⋮

```
In [78]: segment.plot()
```



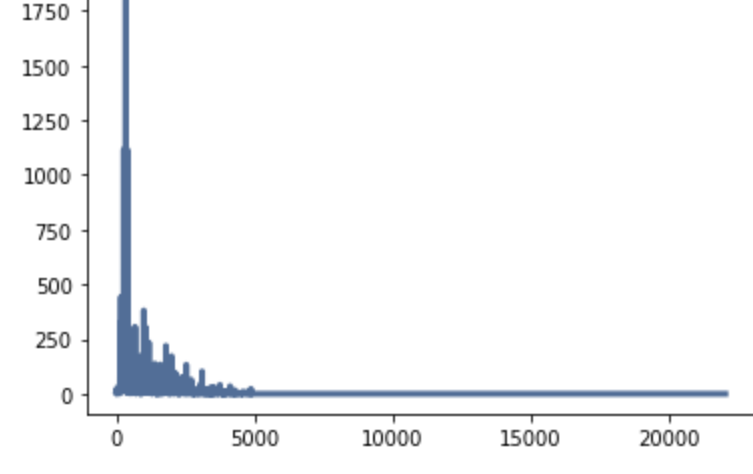
선택한 Segment의 spectrum

```
In [79]: spectrum = segment.make_spectrum()
spectrum.plot()
```



low_pass 사용해보기

```
In [80]: spectrum.low_pass(5000)
spectrum.plot()
```

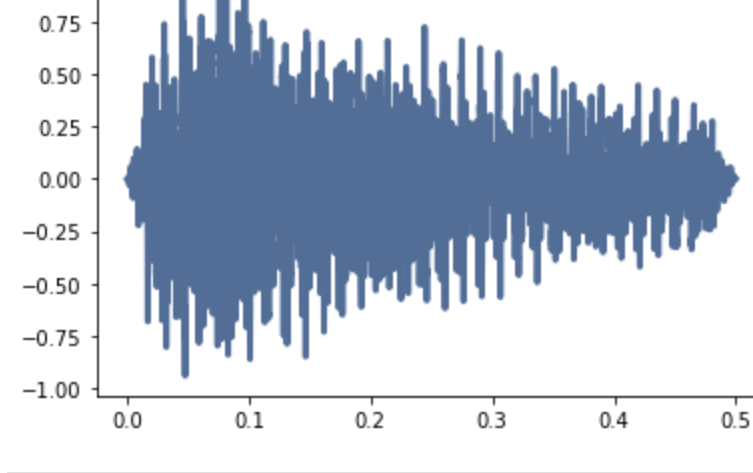


```
In [81]: filtered = spectrum.make_wave()
```

```
In [82]: #정규화 작업
filtered.normalize()
```

부드럽게 앞뒤도 깎아줄, apodize 사용

```
In [83]: filtered.apodize()
filtered.plot()
```



```
In [84]: filtered.make_audio()
```

Out[84]:

▶ 0:00 / 0:00

🔊 ⋮

오리지널 사운드랑 비교 해보기

```
In [85]: segment.make_audio()
```

Out[85]:

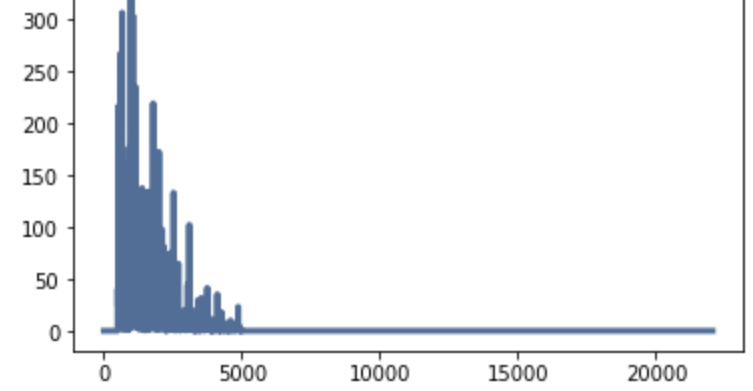
▶ 0:00 / 0:00

🔊 ⋮

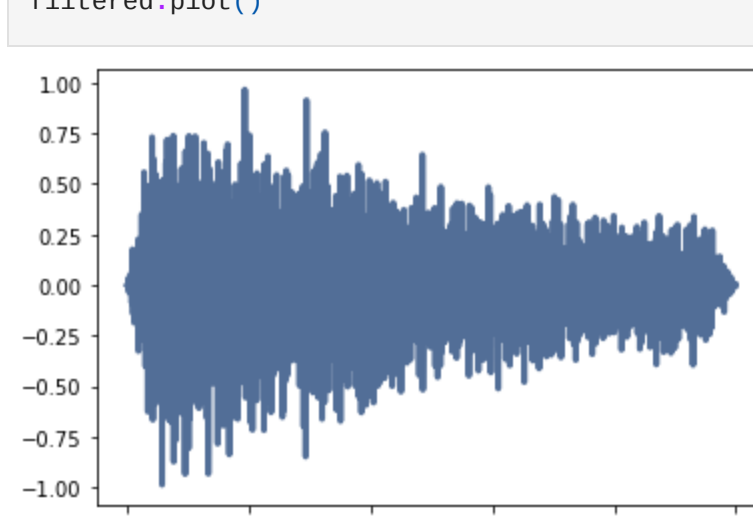
비교결과 음이 오리지널 사운드에 비해, 낮아진것 같습니다.

이번엔 추가로, high_pass 적용해보기

```
In [86]: spectrum.high_pass(500)
spectrum.plot()
```



```
In [87]: filtered = spectrum.make_wave()
filtered.normalize()
filtered.apodize()
filtered.plot()
```



```
In [88]: filtered.make_audio()
```

Out[88]:

▶ 0:00 / 0:00

🔊 ⋮

역시 오리지널 사운드와 비교

```
In [89]: segment.make_audio()
```

Out[89]:

▶ 0:00 / 0:00

🔊 ⋮

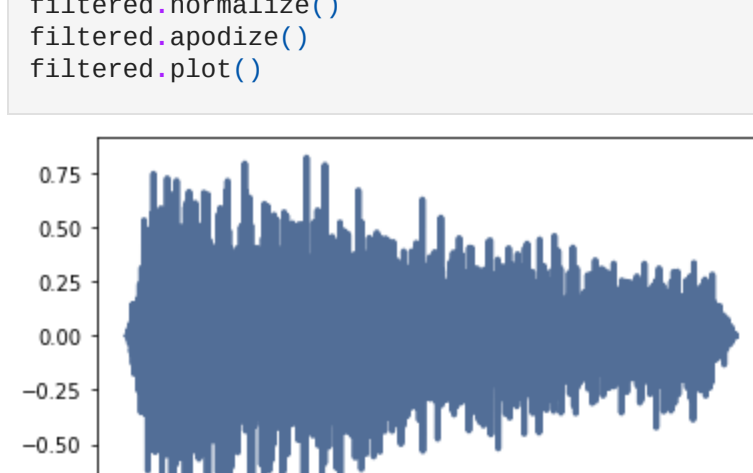
오리지널에 비해, 낮은음들이 안들리는게 확실하게 느껴집니다.

band_stop 사용해보기

```
In [90]: spectrum.band_stop(2000,3000)
spectrum.plot()
```



```
In [91]: filtered = spectrum.make_wave()
filtered.normalize()
filtered.apodize()
filtered.plot()
```



```
In [92]: filtered.make_audio()
```

Out[92]:

▶ 0:00 / 0:00

🔊 ⋮

가운뎃을 비워버려서, 큰 사운드 차이가 날 줄 알았지만, 사운드 차이가 거의 없었습니다.

2000~3000 구간이 분포가 적어서 인 것 같습니다.