ZERO CROSSING RATE

The zero crossing rate is the rate of sign-changes along a signal, i.e., the rate at which the signal changes from positive to negative or back.It usualy has a consideramle diffrence in values for music involving continuous striking of drums etc.. with hands or stick, for instance metal and rock genres

In [1]:
```
import librosa as lr
import librosa.display
import matplotlib.pyplot as plt
import matplotlib.style as ms
%matplotlib inline
import numpy as np
import os
```

Let's analyze a classical audio's zero crossing rate first

In [2]:
```
classical_audio='/home/vivek/Genre_Detection/genres/classical/classical.00000.au'
y , sr =lr.load(classical_audio)
```

In [3]:
```
#the feature returns a numpy array of shape (1,t)
#where element z[0,i] represents the fraction of zero crossing in the i-th frame
zero_crossing_rate=lr.feature.zero_crossing_rate(y)
zero_crossing_rate.shape
```

Out[3]:
```
(1, 1293)
```

Mean of the numpy array thus obtained is taken as a feature for our classification problem

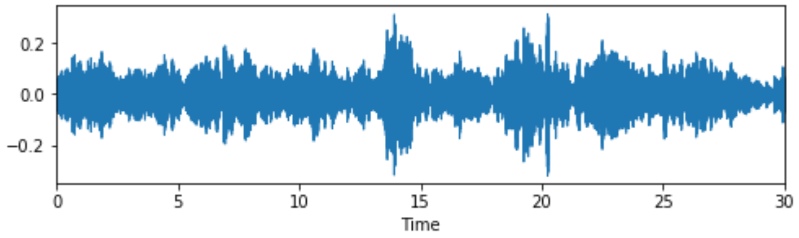
In [4]:
```
np.mean(zero_crossing_rate)
```

Out[4]:
```
0.0982227015661253
```

Visualizing the zero crossing

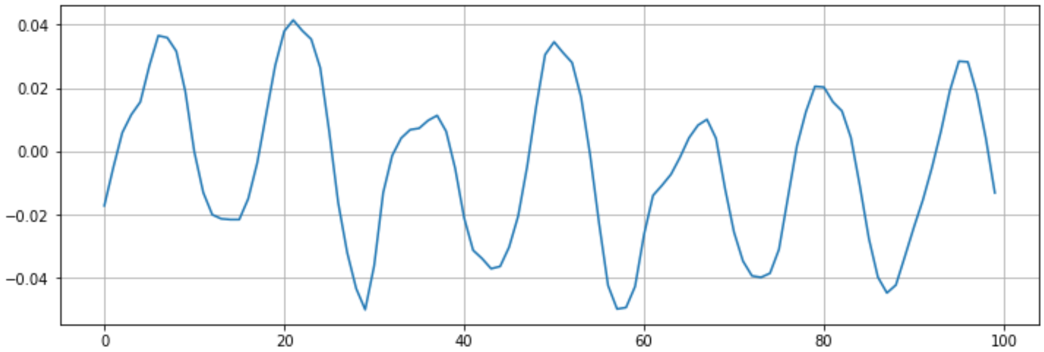
In [5]:
```
plt.figure(figsize=(8, 2))
lr.display.waveplot(y, sr=sr)
```

Out[5]:
```
<matplotlib.collections.PolyCollection at 0x7fbbe27bc3c8>
```



In [6]:

```
# Zooming in
n0 = 1000
n1 = 1100
plt.figure(figsize=(12, 4))
plt.plot(y[n0:n1])
plt.grid()
```



The no of zero crossings is 7 (including the beginning point), checking with librosa

In [7]:

```
zero_crossings=lr.zero_crossings(y[n0:n1])
print(sum(zero_crossings))
```

```
15
```

Now loading a rock audio

In [8]:

```
metal_audio='/home/vivek/Genre_Detection/genres/metal/metal.00001.au'
x , sr1 = lr.load(metal_audio)
```

In [9]:

```
metal_crossing_rate=lr.feature.zero_crossing_rate(x)
np.mean(zero_crossing_rate)
```

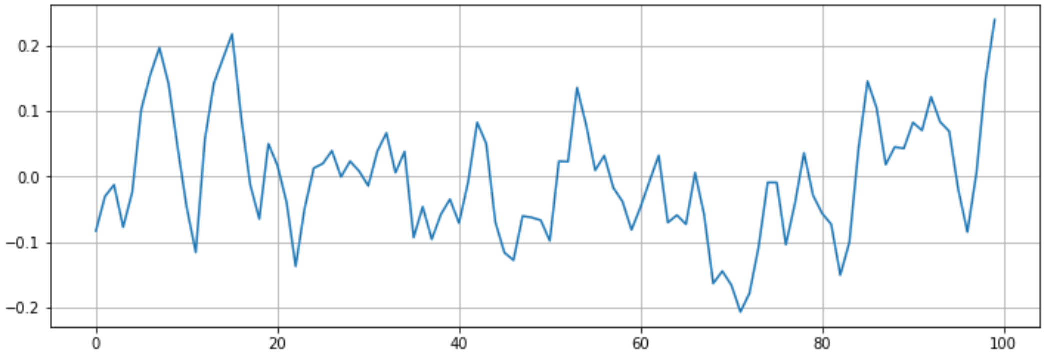
Out[9]:

```
0.0982227015661253
```

It is quite clear that the diffrence is considerable, thus it can be a important feature for classification

In [10]:

```
plt.figure(figsize=(12, 4))
plt.plot(x[n0:n1])
plt.grid()
```



In [11]:

```
zero_crossings=lr.zero_crossings(x[n0:n1])
print(sum(zero_crossings))
```

```
26
```

Zero crossings are significantly larger in genres like metal and rock...(don't compare the beginning and ending of the two genres,the diffrence might not be quite clear)

Sources : **Towards data science (https://towardsdatascience.com/music-genre-classification-with-python-c714d032f0d8)** **Librosa documentation (https://librosa.github.io/librosa/index.html)**

In [ ]:

```
```