

In [0]:

```
#sayısal işaret işleme
#Adem İLHAN-KTU
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
```

"f_s" saniyedeki örnekleme sayısı

"t" zaman aralığı

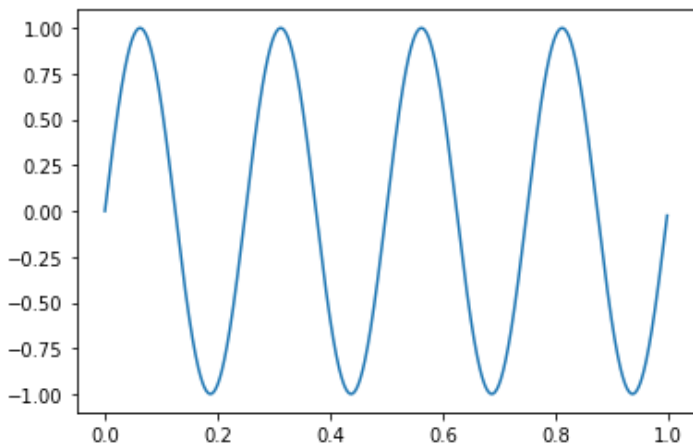
In [0]:

```
f_s = 1000
t = np.array(range(0,1000))/1000.0
#print(t)
```

4 Hz bir sinyalin oluşturulması ve Gösterimi

In [0]:

```
#signal tanımları
signal_4 = np.sin(2*np.pi*4*t) # 4 Hz
plt.plot(t, signal_4)
plt.show()
```



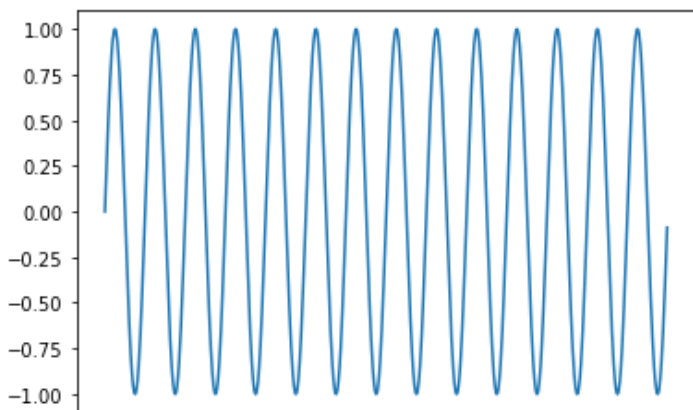
14 Hz bir sinyalin oluşturulması ve Gösterimi

In [0]:

```
signal_14 = np.sin(2*np.pi*14*t) # 14 Hz
plt.plot(t, signal_14)
```

Out[0]:

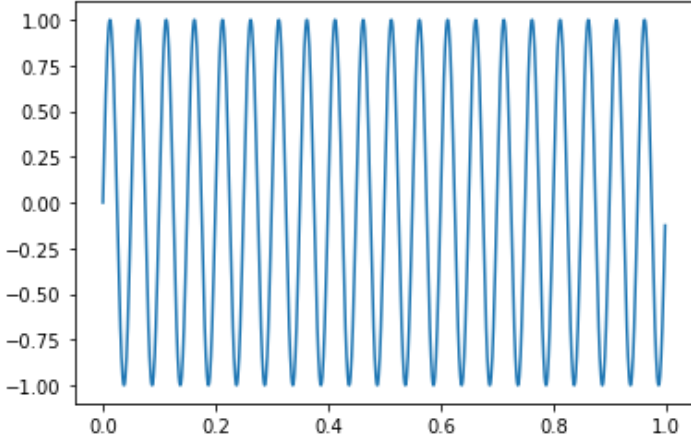
[<matplotlib.lines.Line2D at 0x7fa5846f20f0>]



20 Hz bir sinyalin oluşturulması ve Gösterimi

In [0]:

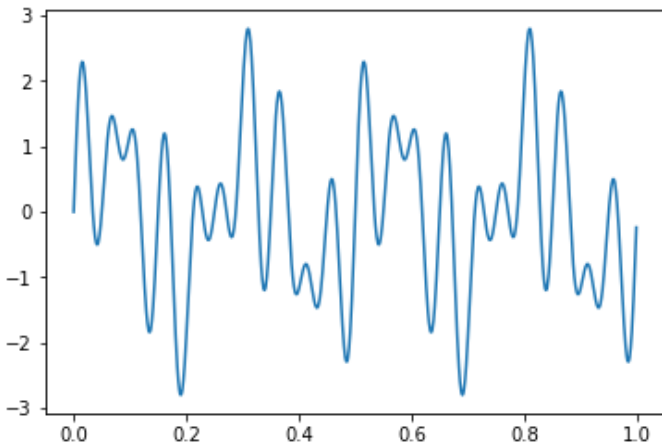
```
signal_20 = np.sin(2*np.pi*20*t) # 20 Hz
plt.plot(t, signal_20)
plt.show()
```



4,14 ve 20 Hz lik sinyaller ile oluşturulan komposit sinyal ve gösterimi

In [0]:

```
komposit = signal_14+signal_20+signal_4
plt.plot(t, komposit)
plt.show()
```



10 Hz lik bir filtrenini oluşturulması. Frekans limitimiz 10 Hz ve örnekleme sıklığımız 1000 olduğu için frekans normalizasyonu için kullanacağımız "w"yi hesaplamak için şu formül kullanılır $w = F_limit/(f_s/2)$. Oluşturulan "w" değeri sonrası Numerator (b) ve denominator (a) değerlerini hesaplıyoruz.

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.butter.html>

Bu işlem sonrası sinyalimizi filtreleyebiliriz. Filtreleme işlemi "filtfilt" methodu ile gerçekleştirilmektedir. Parametre olarak Numerator (b), denominator (a) ve komposit sinyali almaktadır. Dönüş değeri filtrelenmiş sinyaldir. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.filtfilt.html>

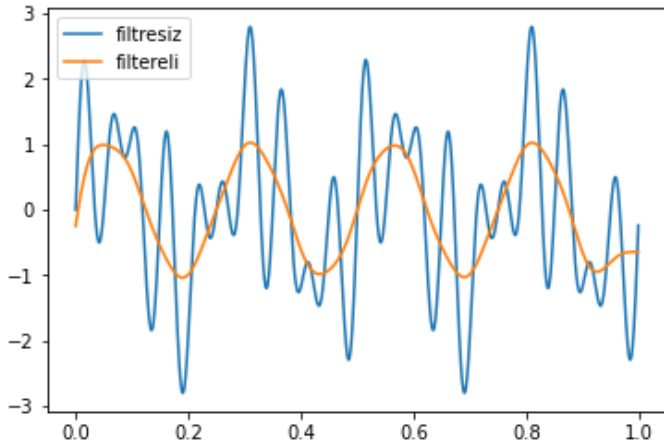
Filtrelenmiş sinyal ile komposit sinyal grafik üzerinde gösterilir.

In [0]:

```
# 10 Hz filter
w = 10 / 500
b, a = signal.butter(5, w, 'low')
output = signal.filtfilt(b, a, komposit)
```

```
plt.plot(t, komposit, label='filtresiz')

plt.plot(t, output, label='filtereli')
plt.legend()
plt.show()
w, h = signal.freqz(output)
#print(h)
#print(len(h))
```

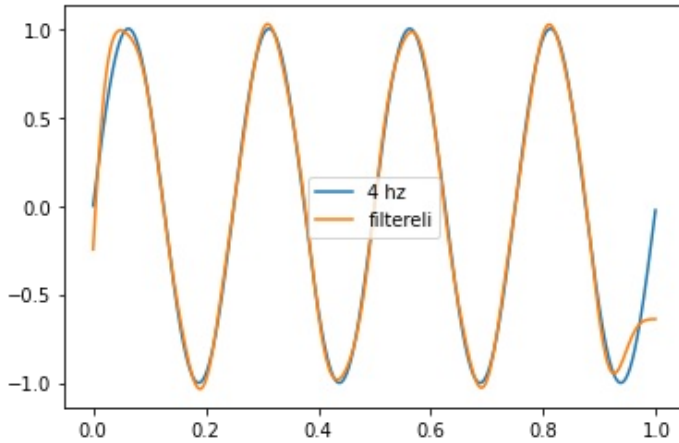


Komposit sinyali incelediğimizde 4,14 ve 20 Hz lik sinyallerden meydana gelmektedir. Bu komposit sinyal 10 Hz lik bir filtreden geçirildiğinde çıkan sonuçun 4 Hz bir sinyal olması beklenmektedir. Bunu doğrulamak için filtrelenmiş sinyal ile 4 Hz lik sinyal aynı düzlem üzerinde gösterilir. İkisi de incelendiğinde 4 Hz'lik sinyaller olduğu söylenebilir.

In [0]:

```
plt.plot(t, signal_4, label='4 hz')

plt.plot(t, output, label='çıkış')
plt.legend()
plt.show()
# büyük oranda doğru
```



In [0]:

Out[0]:

1000

In [0]:

<https://docs.scipy.org/doc/scipy/reference/tutorial/signal.html#fir-filter>