

HALUK TANRIKULU

# PYTHON ile FOURIER DÖNÜŞÜMÜNE GİRİŞ

[www.pitonca.com](http://www.pitonca.com)



# Python ile Fourier Dönüşümüne Giriş

Haluk Tanrikulu

Pitonca | MCNA

Ankara 2019

19 18 17 16 15 14 13 12 11 10

## Yazar tarafından verilen eğitimler:

### Yapay Öğrenme ve Veri Analizi

Python ile Veri Analizine Giriş  
Keras İle Derin Öğrenmeye Giriş  
Makine Öğrenmesine Giriş

### Python Programlama Dili

Python ile Programlama (Giriş, Orta, İleri)  
Python ile Nesne Yönelimli Programlama  
Python ile Soket Programlama  
Matematikçiler için Python Programlama  
Network Mühendisleri için Python  
Python ile Database Programlama  
Python ile DSP'ye Giriş  
Python ile Yapay Zeka Uygulamaları  
Python ile Fourier Dönüşümleri  
Python | Flask ile Programlama  
Python | Django ile Programlama

### HTML5, JavaScript, CSS ile Web Programlama

HTML5 ile Web Tasarımı  
HTML5, JavaScript ve CSS ile Mobile Programlama

### PHP ve MySql İle Programlama

PHP'ye Giriş  
PHP ve MySql ile Web Tasarımı

### Lua Programlama

Lua ile Mobil Oyun Geliştirme  
Mikrokontroller için Lua Programlama

### Ağ Teknolojileri

MCNA Network | Yönlendirme | Ağ Anahtarlama  
Huawei HCNA | HCNF  
Cisco CCNA | CCNP  
Network Fundamentals  
Comptia Network+ | Cloud+

### Siber Güvenlik

Comptia Security+  
Ethical Hacking Eğitimleri (CEH)  
Python ile Güvenlik Uygulamaları Geliştirme  
Sistem Güvenliğine Giriş

### İşletim Sistemleri

Linux (Mint, RedHat, Fedora, Kali, Debian, Ubuntu, SuSE)  
Windows Servers

# Python ile Fourier Dönüşümüne Giriş

Ön Bilgiler



Ver. 1.05 – 2019 Nisan

**Haluk Tanrıkulu**

Tüm Yayın Hakları Haluk Tanrıkulu'na Aittir.

## Fourier Dönüşümün Temelleri

Öncelikle FT Python ile öğrenmek için ön hazırlık yapalım. Yani modülleri yükleyelim.  
Bu bizim birinci adımımız olsun.

```
import numpy as np
import math
import matplotlib.pyplot as plt
import pylab as pl
from IPython import display
import time as ttime
import random
from mpl_toolkits.mplot3d import Axes3D
```

Anaconda ile bu modülleri yüklemiş olalım.

Öncelikle FT Python ile öğrenmek için ön hazırlık yapalım. Yani modülleri

## Kompleks Sayılar (Complex Numbers)

**Kompleks sayılar önemli önce ona bakalım.**

```
# kompleks sayılar real + imaginary şeklinde yazılır.
z = 4 + 3j
print(z)
# veya
z = 4 + 1j * 3
print(z)
```

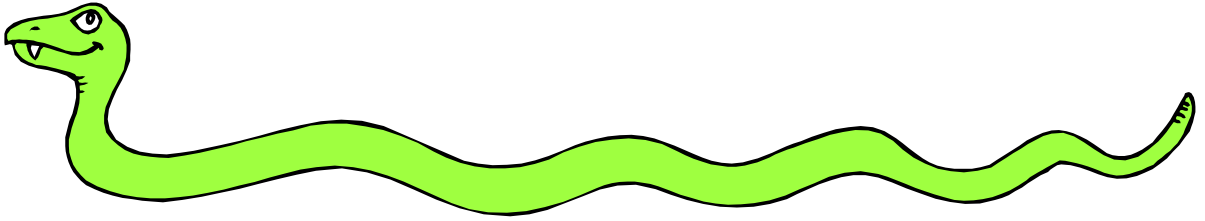
```
# komplex (complex) fonksiyonunu kullanalım
```

```
z = complex(4,3) # ya böyle
print(z)
z = complex('4+3j') # yada böyle
```

Çalıştıralım.

```
(4+3j)
(4+3j)
(4+3j)
(4+3j)
```

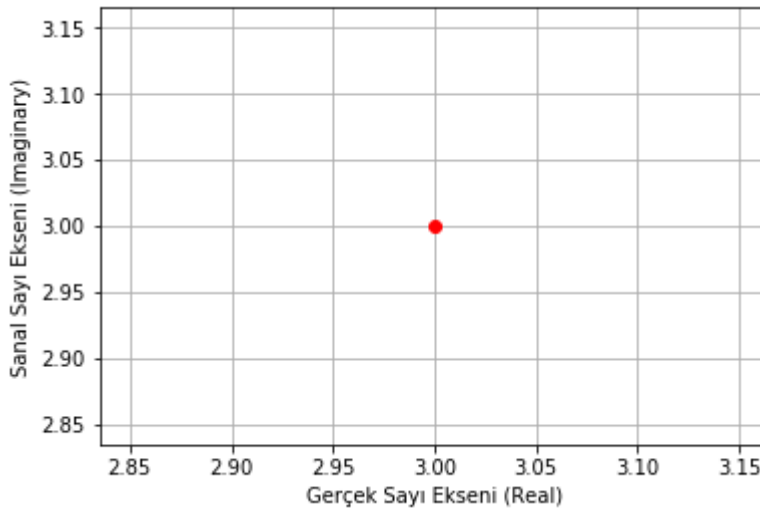
Çıkar.. nasıl yazarsak yazalım aynı sonuç çıktı...



## Komplex Sayının Gösterimi

# Komplex koordinat sisteminde bir kompleks sayıyı gösterelim.

```
z = 3 + 3j
plt.plot(np.real(z), np.imag(z), 'ro') # kırmızı o
plt.grid(True)
plt.xlabel('Gerçek Sayı Eksen (Real)'), plt.ylabel('Sanal Sayı Eksen (Imaginary)')
plt.show()
```



# çizimde bazı düzenlemeler yapalım.

$z = 3 + 3j$

```
plt.plot(np.real(z),np.imag(z),'ro') # kırmızı o
```

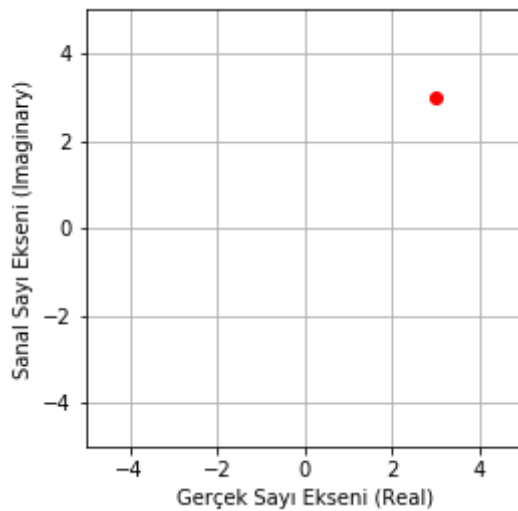
```
plt.axis('square')
```

```
plt.axis([-5, 5, -5, 5])
```

```
plt.grid(True)
```

```
plt.xlabel('Gerçek Sayı Eksen (Real)'), plt.ylabel('Sanal Sayı Eksen (Imaginary)')
```

```
plt.show()
```



## Magnitue (Mutlak Değerin Bulunması)

# Pisagor Teoreminden buluruz.

```
mag = np.sqrt( np.real(z)**2 + np.imag(z)**2 )
```

```
print( 'Magnitude Değeri : ',mag )
```

# yada doğrudan abs() fonksiyonu kullanırız.

```
mag = np.abs(z)
```

```
print( 'Magnitude Değeri : ',mag )
```

Sonuç ne çıkar... Bakalım:

Magnitude Değeri : 4.242640687119285

Magnitude Değeri : 4.242640687119285

## Açı Değerinin Bulunması

# arctanjant kullanalım.

```
mag = math.atan( np.imag(z) / np.real(z) )
```

```
print( 'Açı Değeri : ',phs )
```

# yada angle() fonksiyonu ile yapalım.

```
phs = np.angle(z)
```

```
print( 'Açı Değeri : ',phs )
```

Sonuç nedir?

Açı Değeri : 0.7853981633974483

Açı Değeri : 0.7853981633974483

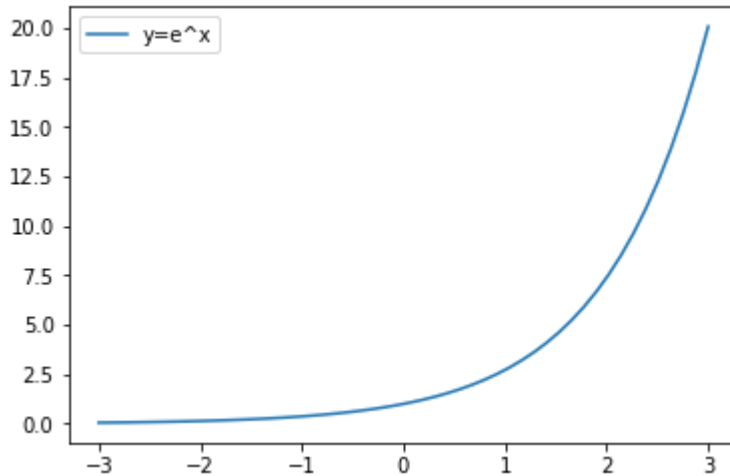
## e ile Euler Formülüne Bakalım

```
x = np.linspace(-3,3,num=50)
```

```
plt.plot(x,np.exp(x),label='y=e^x')
```

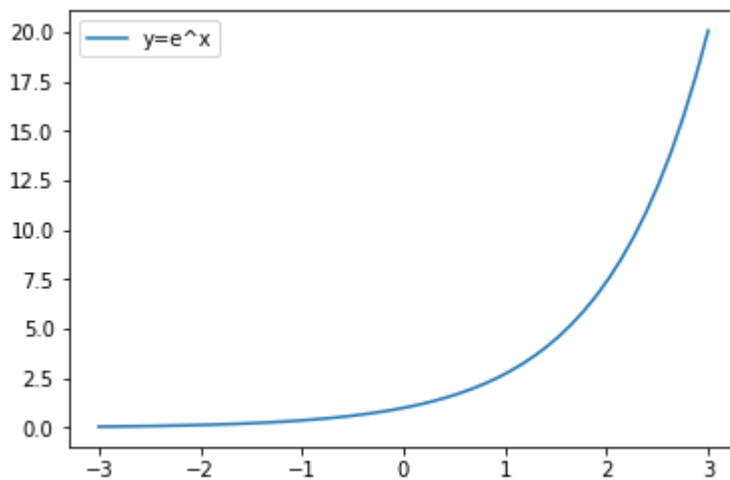
```
plt.show()
```



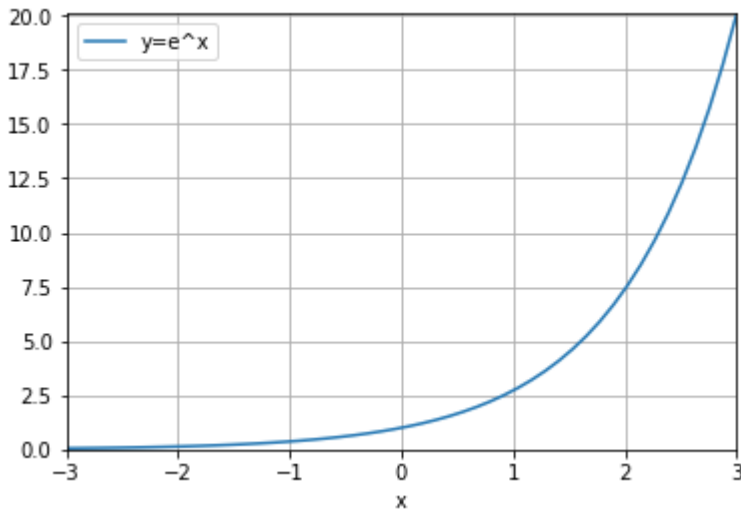


Bunu biraz değiştirelim. Bildiğimiz gibi yazalım.

```
plt.plot(x,np.e**(x),label='y=e^x')  
plt.legend()  
plt.show()
```



```
# Biraz düzeltme yapalım.  
plt.plot(x,np.exp(x),label='y=e^x')  
plt.axis([min(x),max(x),0,np.exp(x[-1])])  
plt.grid(True)  
plt.legend()  
plt.xlabel('x')  
plt.show()
```



Şimdi Birim Çember üzerinde herhangi bir gerçel sayı olan  $k$  için  $[\cos(k), \sin(k)]$  değerlerini çizelim.

```
# Bir k değeri tanımlayalım.
```

```
k = 2/np.pi
```

```
# Euler gösterimi şu şekilde olsun.
```

```
euler = np.exp(1j*k)          #yani euler = e^jk
```

```
# Bulunduğu noktayı çizelim.
```

```
plt.plot(np.cos(k),np.sin(k),'ro')
```

```
# referans birim çemberi çizelim.
```

```
x = np.linspace(-np.pi,np.pi,num=100) #
```

```
plt.plot(np.cos(x),np.sin(x))
```

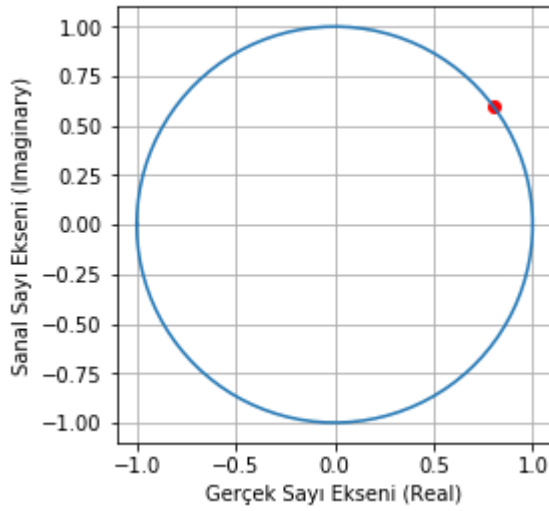
```
# biraz güzelleştirelim
```

```
plt.axis('square')
```

```
plt.grid(True)
```

```
plt.xlabel('Gerçek Sayı Eksen (Real)'), plt.ylabel('Sanal Sayı Eksen (Imaginary)')
```

```
plt.show()
```

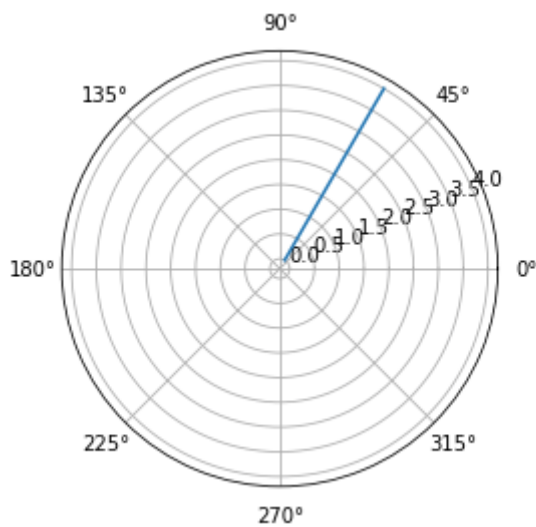


# Euler formülü keyfi vektör büyüklüğü ile

```
m = 4          # genlik
k = np.pi/3    # açı
compnum = m*np.exp( 1j*k )
```

```
# extract magnitude and angle
mag = np.abs(compnum)
phs = np.angle(compnum)
```

```
plt.polar([phs,phs],[0,mag])
plt.show()
```



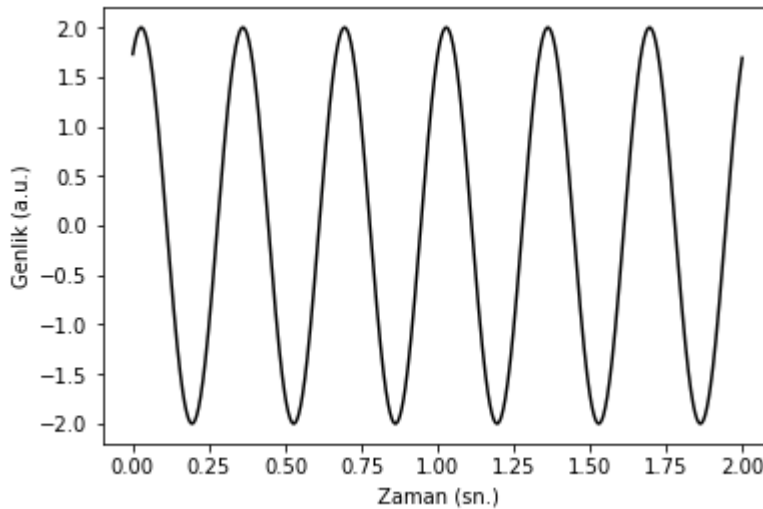
## Sinüs dalgası ve kompleks sinüs dalgası

```
# Benzetim parametreleri
srate = 500;                # örnekleme frekansı (Hz)
time = np.arange(0.,2.,1./srate) # zaman (sn)

# Sinüs dalgası parametreleri
for i in range(1, 4):

    freq = i                # frekans in Hz
    ampl = 2                # genlik a.u.
    phas = np.pi/3        # faz radyan
    freq = 3
# Sinüs dalgası oluşturalım
sinewave = ampl * np.sin( 2*np.pi * freq * time + phas )

plt.plot(time,sinewave,'k')
plt.xlabel('Time (sec.)')
plt.ylabel('Amplitude (a.u.)')
plt.show()
```



Bu yayının tüm hakları saklıdır. Hiçbir şekilde kopyalanamaz, çoğaltılamaz.