# Signals and Systems Theory (COMM401)

## Lab 2

## Signal representation

**Prepared by:** Eng. Youstina Megalli
**Supervised By:** Prof. Dr. Ahmed El-Mahdy

German University in Cairo
الجامعة الألمانية بالقاهرة

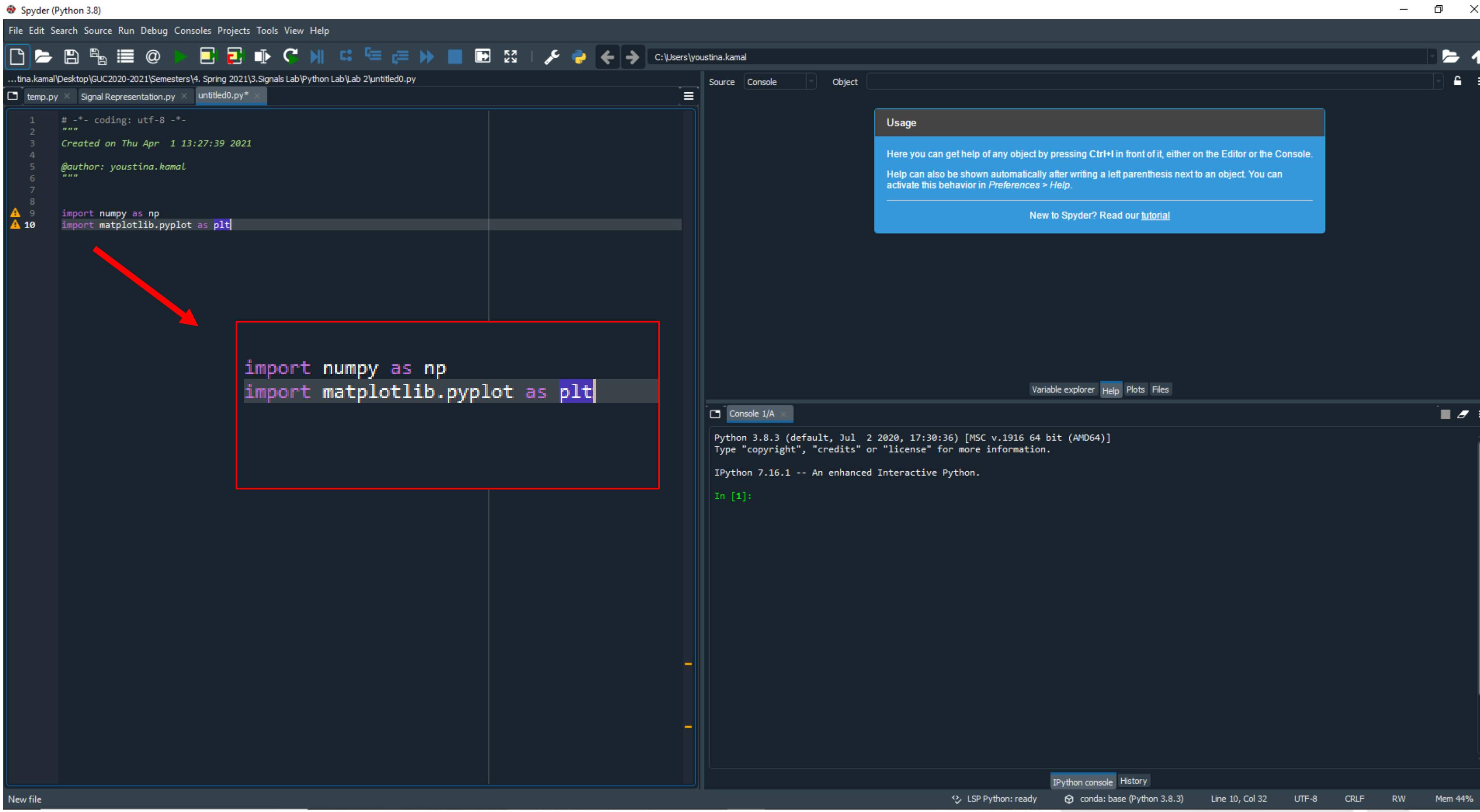
Faculty of Information Engineering and Technology
Department of Communication Engineering

# Python: Programming Language

- Platform: ANACONDA
- Environment: SPYDER

# Importing Libraries

**Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy.**
**As such, it offers a viable open source alternative to MATLAB.**




GUC
German University in Cairo
الجامعة الألمانية بالقاهرة

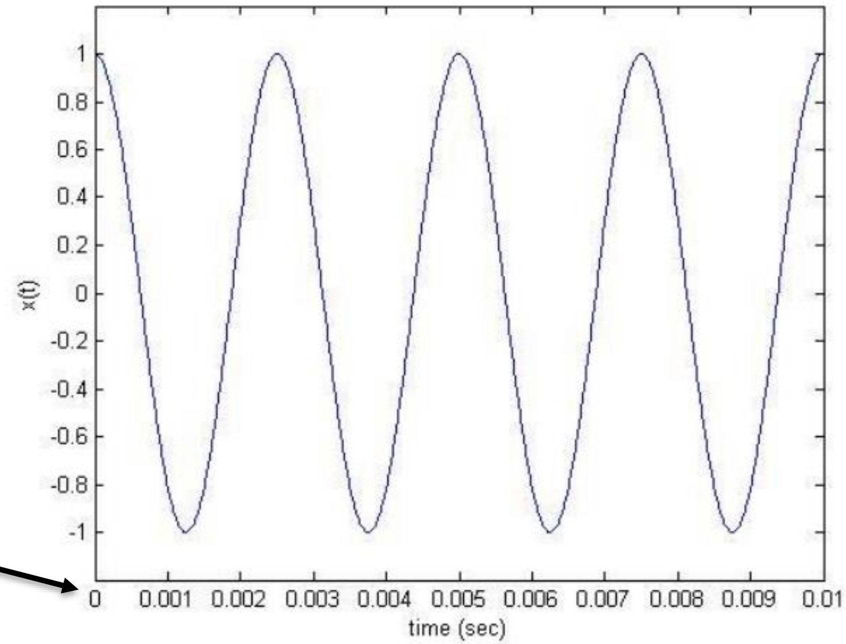
# What is a signal?

It is a function representing a physical quantity. Mathematically, signals are represented as functions of one or more independent variables.

**Example:**

sinusoidal signal $x(t) = \cos(t)$

Function of one independent variable t

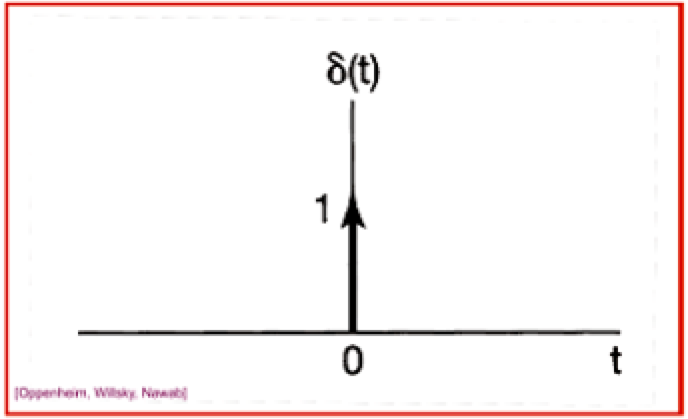



German University in Cairo
الجامعة الألمانية بالقاهرة

# Famous Signals

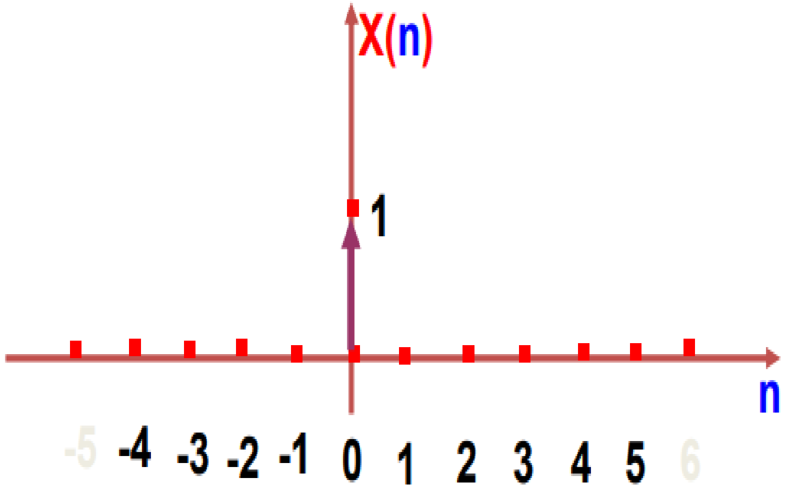
- **Unit Impulse Function:**

## Continuous Time




## Discrete Time

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

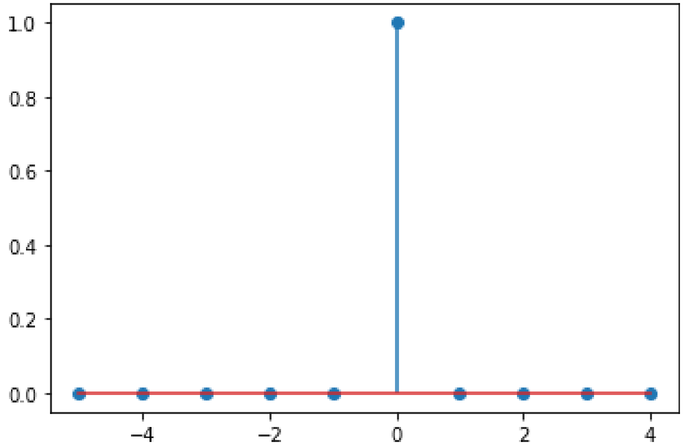


# Plotting

- Plotting a discrete signal:
  **matplotlib.pyplot.stem**(*args, linefmt=None, markerfmt=None, basefmt=None,bottom=0, label=None, use_line_collection=True, orientation='vertical',data=None)

- Examples : Plotting a dirac function

```
n= np.arange(-5,5)
x=np.reshape([n==0],np.shape(n))
plt.stem(n,x,use_line_collection='True')
```



# Plotting

- Plotting a discrete signal:
  **matplotlib.pyplot.stem**(*args, linefmt=None, markerfmt=None, basefmt=None,bottom=0, label=None, use_line_collection=True, orientation='vertical')

- Linefmt → str, optional

A string defining the color and/or linestyle of the vertical lines

- Markerfmt → str, optional

A string defining the color and/or shape of the markers at the stem heads.

- basefmtstr → default: 'C3-' ('C2-' in classic mode)

A format string defining the properties of the baseline.

- Bottomfloat → default: 0

The y/x-position of the baseline (depending on orientation).

- use_line_collectionbool → default: True

If True, store and plot the stem lines as a LineCollection instead of individual lines, which significantly increases performance

# Plotting

```
n= np.arange(-5,5)
x=np.reshape([n==0],np.shape(n))
plt.stem(n,x,use_line_collection='True')
```

n → array of integers

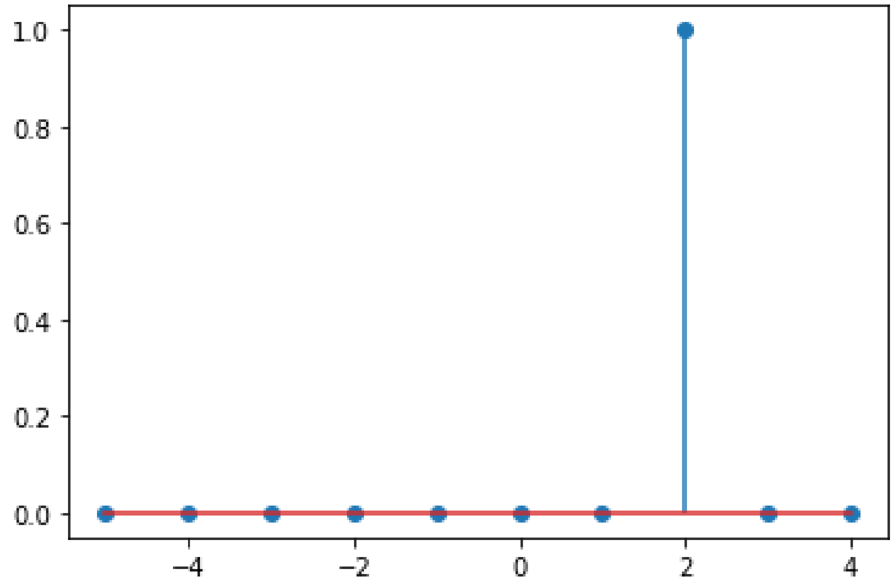
n==0 → List

np.reshape([n==0],np.shape(n)) → Array of Booleans of same size of n

# Plotting

Plotting a shifted dirac

```
n= np.arange(-5,5)
x=np.reshape([(n-2)==0],np.shape(n))
plt.stem(n,x,use_line_collection='True')
```



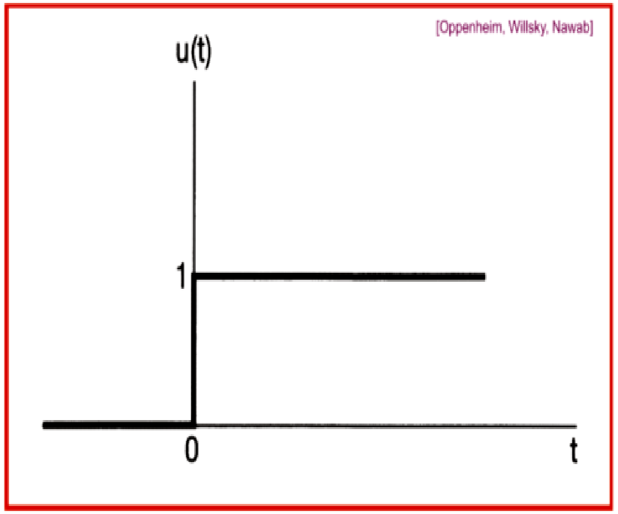
# Famous Signals

- **Unit Step Function:**

## Continuous Time

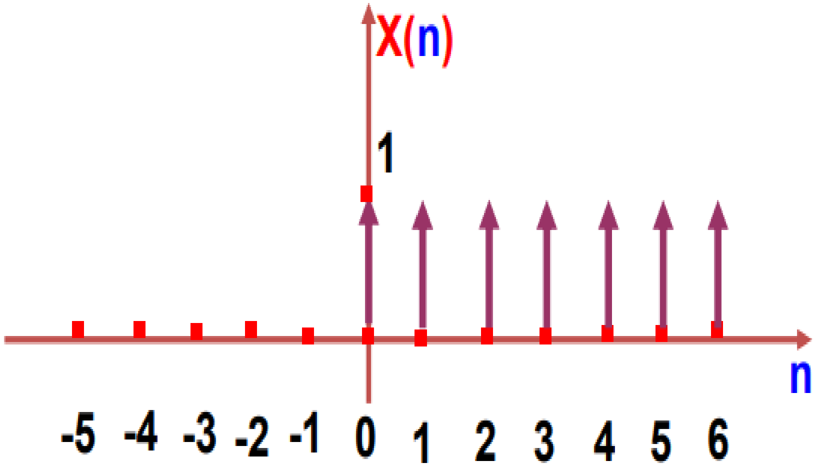
$$u(t) = \begin{cases} 0, & t < 0 \\ 1, & t > 0 \end{cases}$$

Note that the unit step function is discontinuous at $t = 0$.




## Discrete Time

$$U(n) = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$

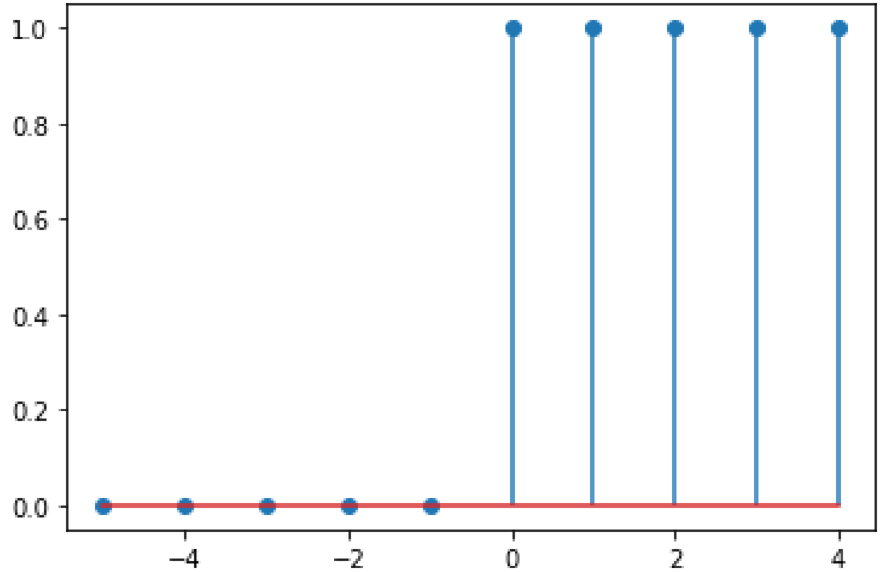



German University in Cairo
الجامعة الألمانية بالقاهرة

# Plotting

Plotting a step function

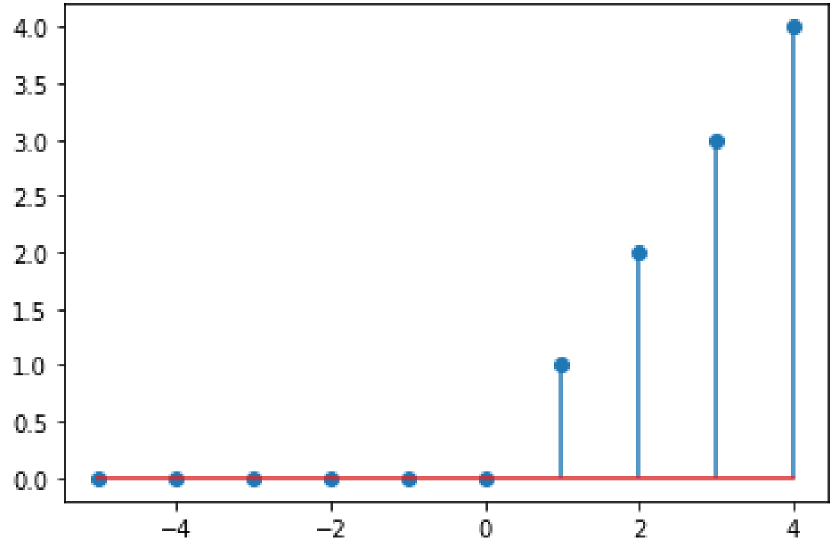
```
n= np.arange(-5,5)
x=np.reshape([n>=0],np.shape(n))
plt.stem(n,x,use_line_collection='True')
```



# Plotting

Plotting an increasing function

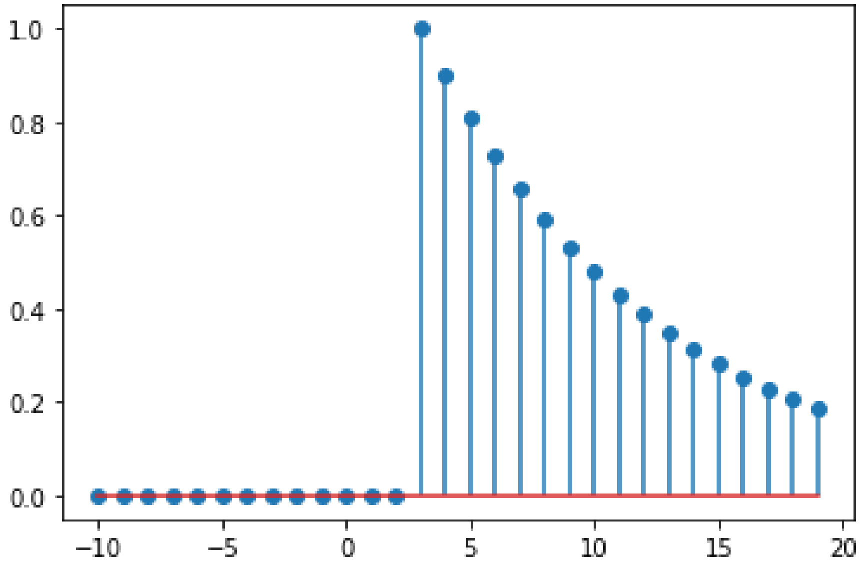
```
n= np.arange(-5,5)
x=np.reshape(n*[n>=0],np.shape(n))
plt.stem(n,x,use_line_collection='True')
```



# Plotting

Plotting a decreasing function

```
n= np.arange(-10,20)
x=np.reshape((0.9**(n-3))*[(n-3)>=0],np.shape(n))
plt.stem(n,x,use_line_collection='True')
```



# Mathematical Review

- **Complex Numbers:**

$$z = x + jy = r\angle\theta$$

where

$\boldsymbol{x}$ is the real part

$\boldsymbol{y}$ is the imaginary part

$\boldsymbol{r}$ is the magnitude. $r = \sqrt{x^2 + y^2}$

$\boldsymbol{\theta}$ is the angle. $\theta = \tan^{-1}\frac{y}{x}$

- **Complex exponentials:**

$$e^{j\omega_0 t} = \cos(\omega_0 t) + j\sin(\omega_0 t)$$

$$\left|e^{j\omega_0 t}\right| = \sqrt{(\cos(\omega_0 t))^2 + (\sin(\omega_0 t))^2}$$

$$\boxed{\therefore \left|e^{j\omega_0 t}\right| = 1}$$

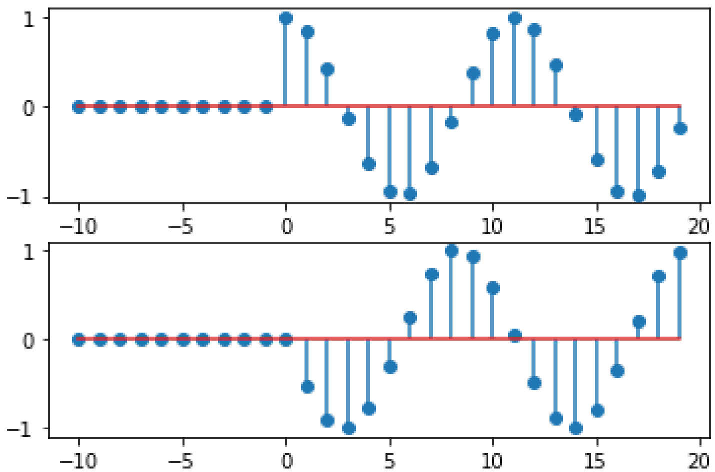
German University in Cairo

الجامعة الألمانية بالقاهرة

# Plotting

Plotting real and imaginary parts of a signal

```
n= np.arange(-10,20)
x=np.reshape(np.exp(12j*n)*[n>=0],np.shape(n))
# plt.stem(n,x,use_line_collection='True')
y=np.real(x)
z=np.imag(x)
plt.subplot(2,1,1)
plt.stem(n,y ,use_line_collection='True')
plt.subplot(2,1,2)
plt.stem(n,z, use_line_collection='True')
```



# Plotting

Plotting real and imaginary parts of a signal