

4.5. Matplotlib

January 26, 2018

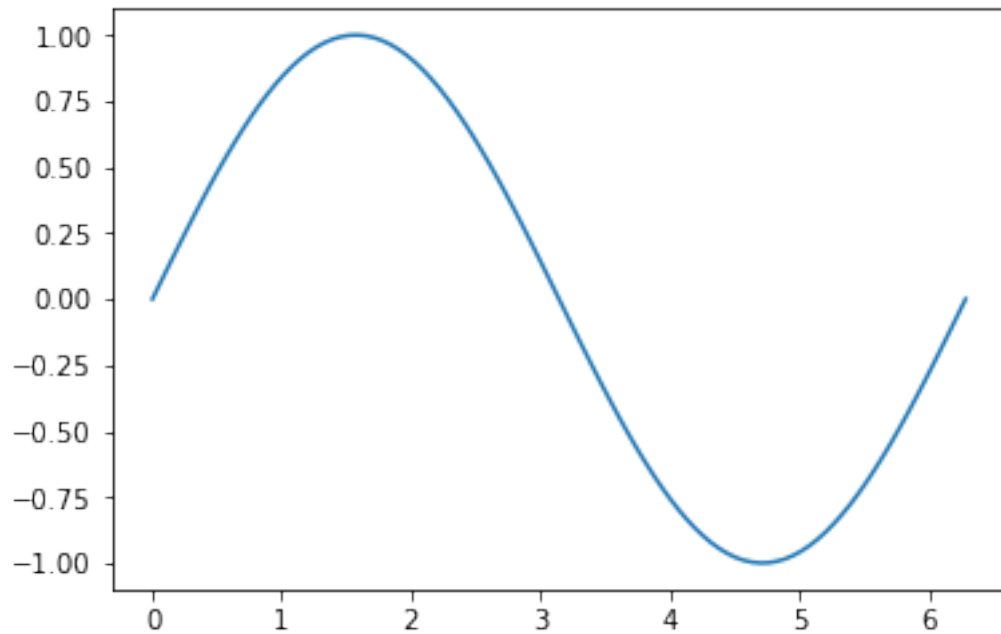
1 Matplotlib

Se presentan todos los ejercicios vistos en el documento teorico.

1.1 1. Gráficas Simples

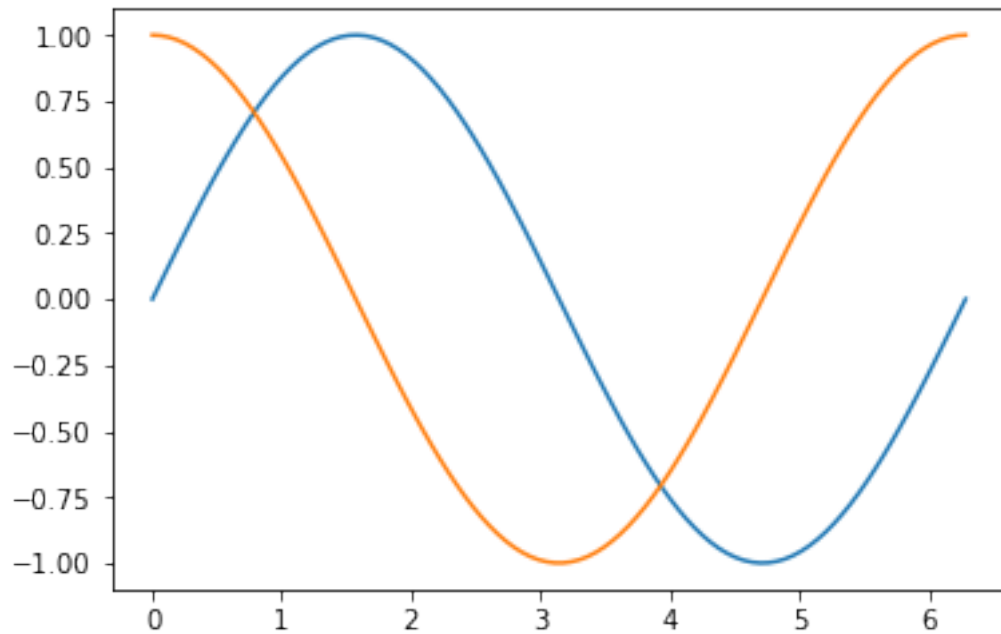
1.1.1 Dibujar la función senoidal para 100 puntos en el intervalo de 0 a 2π .

```
In [2]: import numpy as np # Importamos el módulo numpy
import matplotlib.pyplot as plt # Importamos el modulo pyplot de numpy
# Generamos un arreglo de 100
# puntos que va desde 0 a  $2\pi$ 
x = np.linspace(0,2*np.pi,100) # Dominio
# Evaluamos cada punto del dominio
# en la funcion senoidal y lo almacenamos
y = np.sin(x) # Rango
# Graficamos el resultado
plt.plot(x,y)
# Mostramos en pantalla
plt.show()
```



1.1.2 Dibujamos la función senoidal y cosenoidal en la misma ventana.

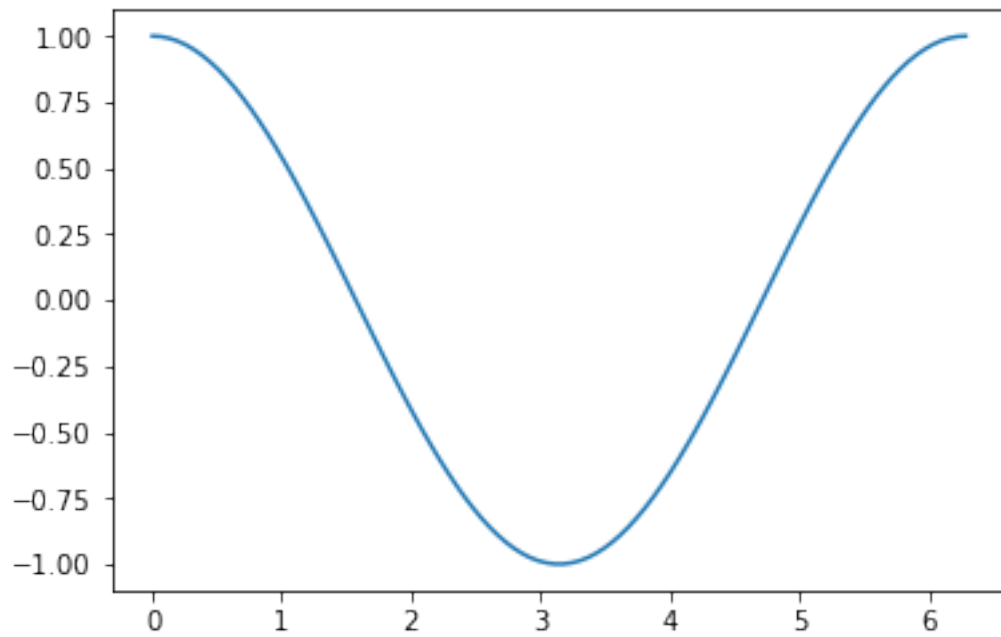
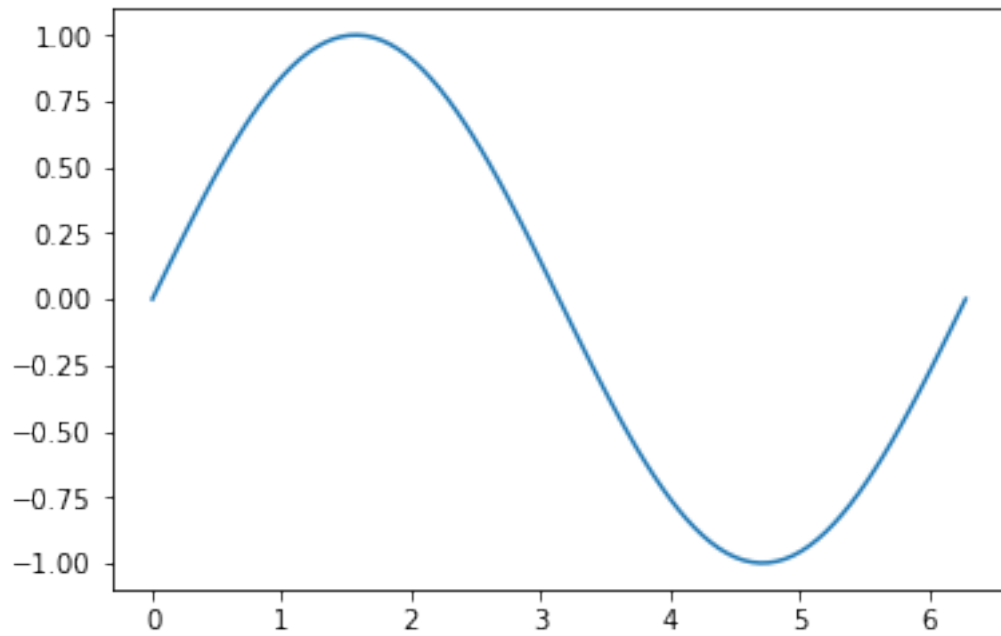
```
In [3]: import numpy as np # Importamos el módulo numpy
import matplotlib.pyplot as plt # Importamos el modulo pyplot de numpy
# Generamos un arreglo de 100
# puntos que va desde 0 a 2pi
x = np.linspace(0,2*np.pi,100) # Dominio
# Evaluamos cada punto del dominio
# en la funcion senoidal y lo almacenamos
y = np.sin(x) # Rango
# Evaluamos cada punto del dominio
# en la funcion cosenoidal y lo almacenamos
f = np.cos(x) # Rango 2
# Graficamos el resultado
plt.plot(x,y) # Graficamos la función senoidal
plt.plot(x,f) # Graficamos la función cosenoidal
# Mostramos en pantalla
plt.show()
```



1.1.3 Dibujamos en pantalla la función cosenoidal y senoidal en distintas figuras.

```
In [1]: import numpy as np
import matplotlib.pyplot as plt

# Creamos dos figuras distintas en donde graficar
plt.figure(1) # Identificador 1
plt.figure('dos') # Identificador dos
# Generamos el dominio de 0 a 2pi con 100 puntos
x = np.linspace(0,2*np.pi,100)
# Hacemos la evaluacion de estos puntos
y = np.sin(x)
f = np.cos(x)
# Dibujamos la funcion coseno en la ventada dos
plt.figure('dos')
plt.plot(x,f)
# Dibujamos la funcion seno en grafica 1
plt.figure(1)
plt.plot(x,y)
# Mostramos en pantalla el resultado
plt.show()
```



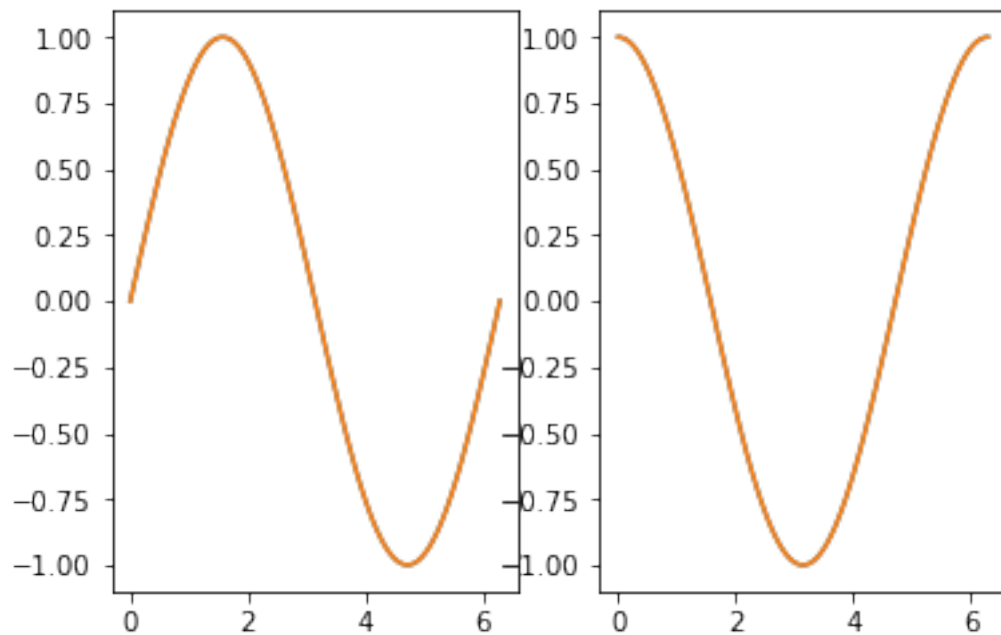
1.1.4 Dibuja la función senoidal y cosenoidal en la misma ventana.

```
In [10]: import numpy as np  
         import matplotlib.pyplot as plt
```

```

# Generamos el dominio de 0 a 2pi con 100 puntos
x = np.linspace(0,2*np.pi,100)
# Hacemos la evaluacion de estos puntos
y = np.sin(x)
f = np.cos(x)
# Dibujamos la funcion coseno en la primera
# mitad de la grafica
plt.subplot(1,2,1)
plt.plot(x,y)
# Dibujamos la funcion seno en la segunda
# mitad de la grafica
plt.subplot(1,2,2)
plt.plot(x,f)
plt.show()

```



1.2 Personalización

Ejemplos sobre la personalización de los gráficos, para que el usuario pueda presentar con mayor claridad sus resultados.

1.3 Mostrar en pantalla la función senoidal y cosenoidal con leyendas y una malla.

```

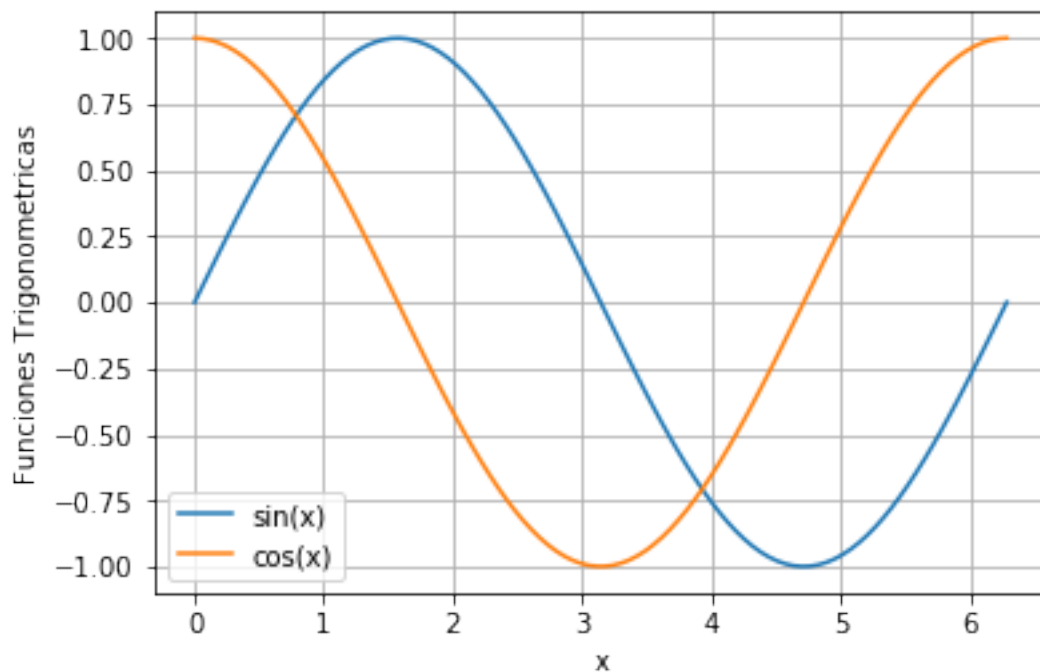
In [11]: import numpy as np
import matplotlib.pyplot as plt
# Generamos el dominio

```

```

x = np.linspace(0,2*np.pi,100)
# Mostramos en pantalla la grafica del seno
plt.plot(x,np.sin(x),label="sin(x)")
# Mostramos en pantalla la grafica de coseno
plt.plot(x,np.cos(x),label="cos(x)")
# Agregamos las leyendas
plt.legend()
# Agregamos una malla al grafico
plt.grid()
# Etiqueta a los ejes
plt.xlabel("x")
plt.ylabel("Funciones Trigonometricas")
plt.show()

```



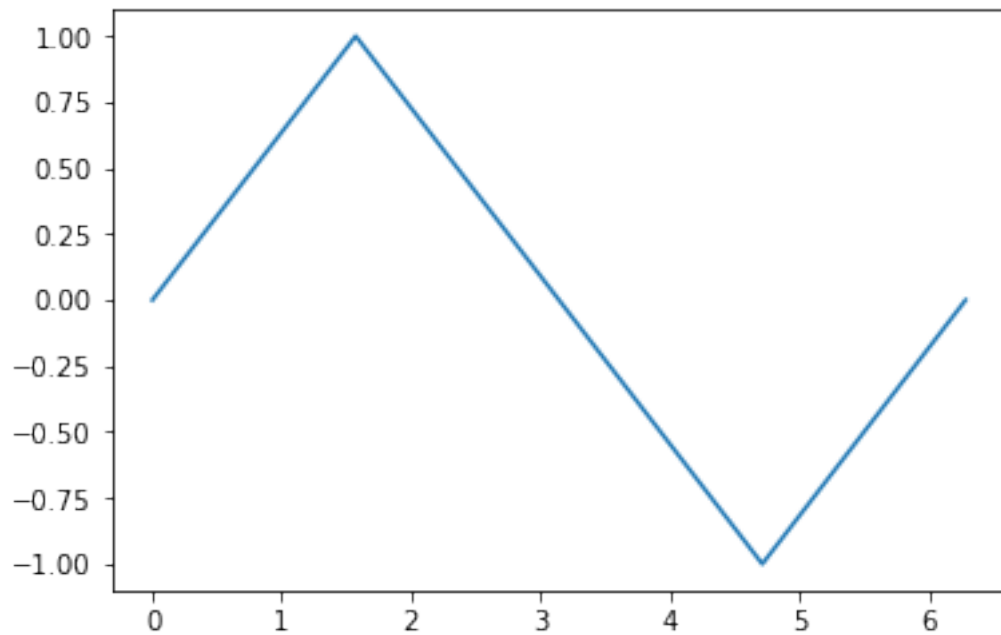
1.4 Gráficar la función senoidal de 0 a 2pi con 5 puntos.

```

In [2]: # Agregamos la libreria de numpy
import numpy as np
# Agregamos la libreria matplotlib
import matplotlib.pyplot as plt
# Generamos el dominio con linspace
x = np.linspace(0,2*np.pi,5)
# Graficamos la funcion
plt.plot(x,np.sin(x))

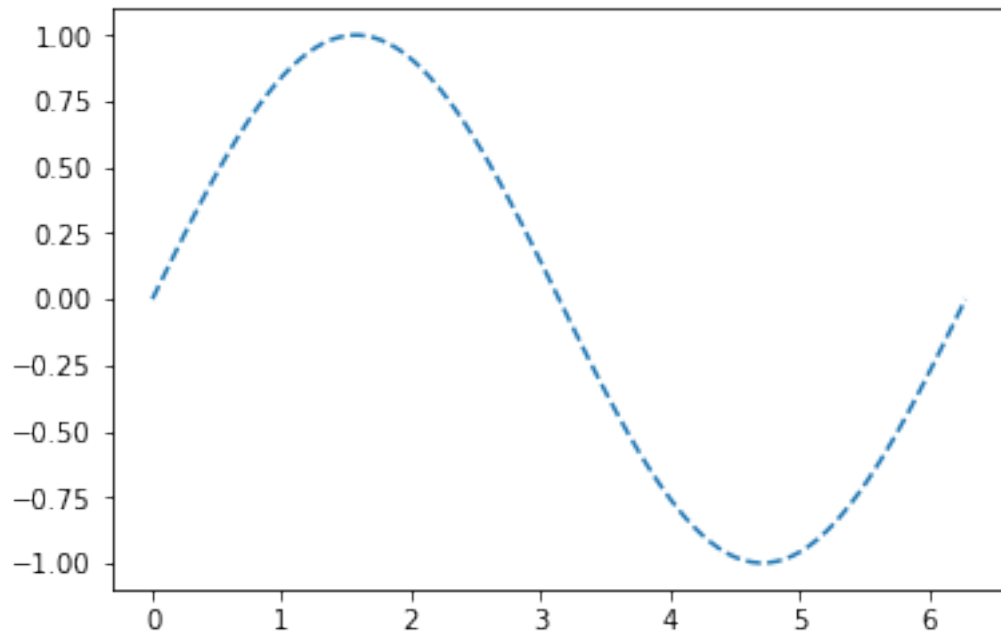
```

```
# Mostramos en pantalla  
plt.show()
```



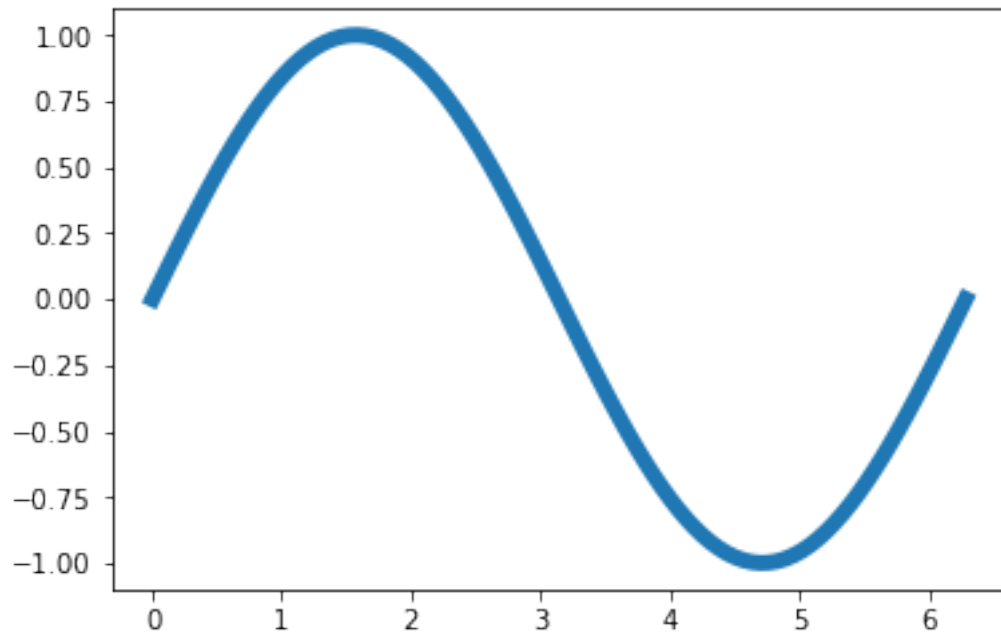
1.5 Gráficar la función senoidal de 0 a 2π con 100 puntos y la línea discontinua.

```
In [16]: # Agregamos la libreria de numpy  
import numpy as np  
# Agregamos la libreria matplotlib  
import matplotlib.pyplot as plt  
# Generamos el dominio con linspace  
x = np.linspace(0, 2*np.pi, 100)  
# Graficamos la funcion  
plt.plot(x, np.sin(x), linestyle='dashed')  
# Mostramos en pantalla  
plt.show()
```



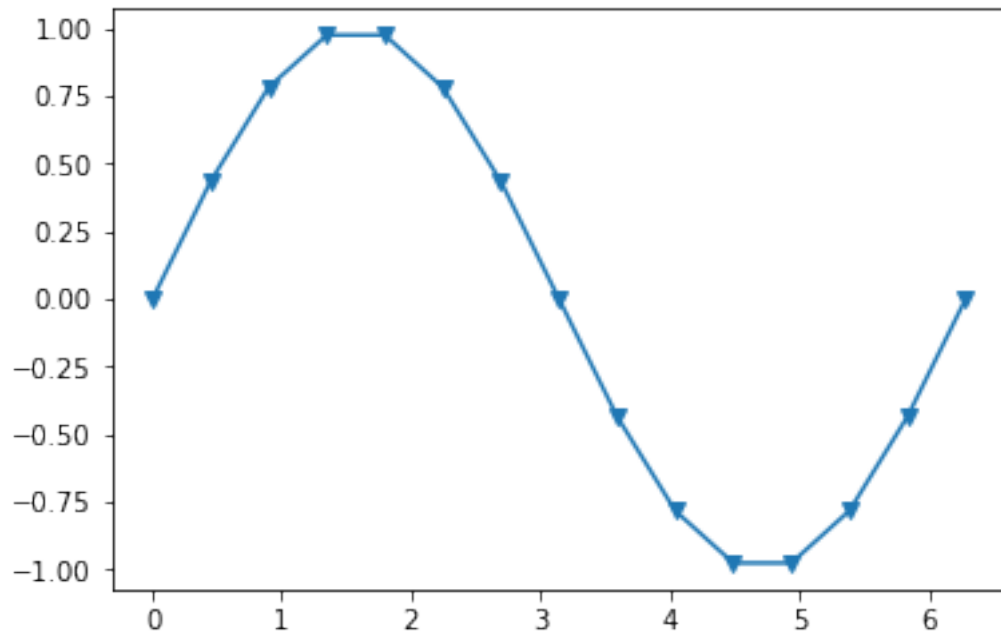
1.6 Gráficar la función senoidal de 0 a 2π con 100 puntos y la línea con un espesor mayor.

```
In [17]: # Agregamos la libreria de numpy
import numpy as np
# Agregamos la libreria matplotlib
import matplotlib.pyplot as plt
# Generamos el dominio con linspace
x = np.linspace(0,2*np.pi,100)
# Graficamos la funcion
plt.plot(x,np.sin(x),linewidth=6)
# Mostramos en pantalla
plt.show()
```

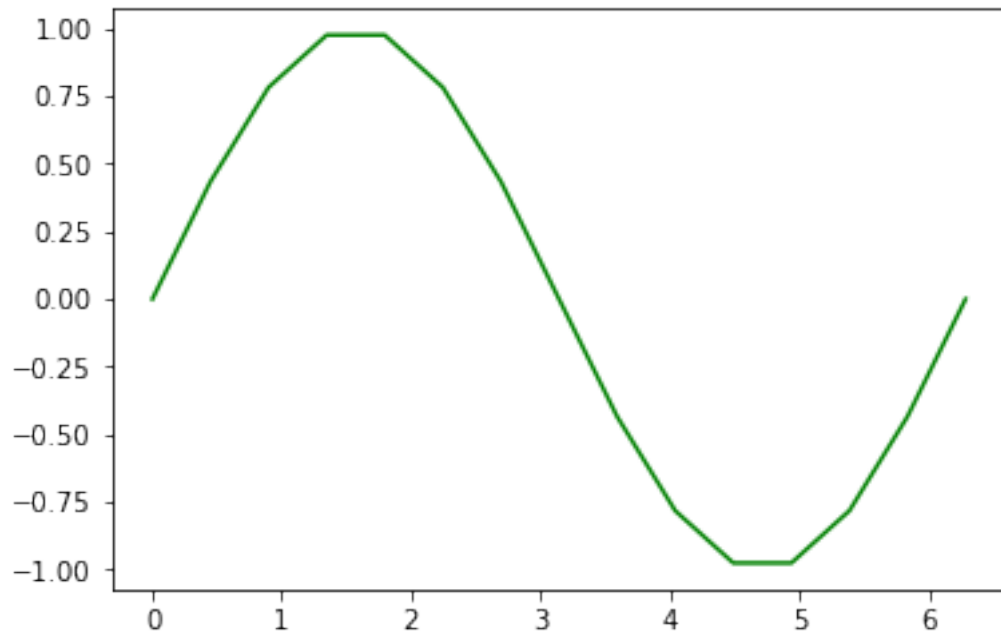
1.7 Gráficar la función senoidal de 0 a 2π con 15 puntos y un marcador triangular.

```
In [22]: # Agregamos la libreria de numpy
import numpy as np
# Agregamos la libreria matplotlib
import matplotlib.pyplot as plt
# Generamos el dominio con linspace
x = np.linspace(0,2*np.pi,15)
# Graficamos la funcion
plt.plot(x,np.sin(x),marker='v')
# Mostramos en pantalla
plt.show()
```



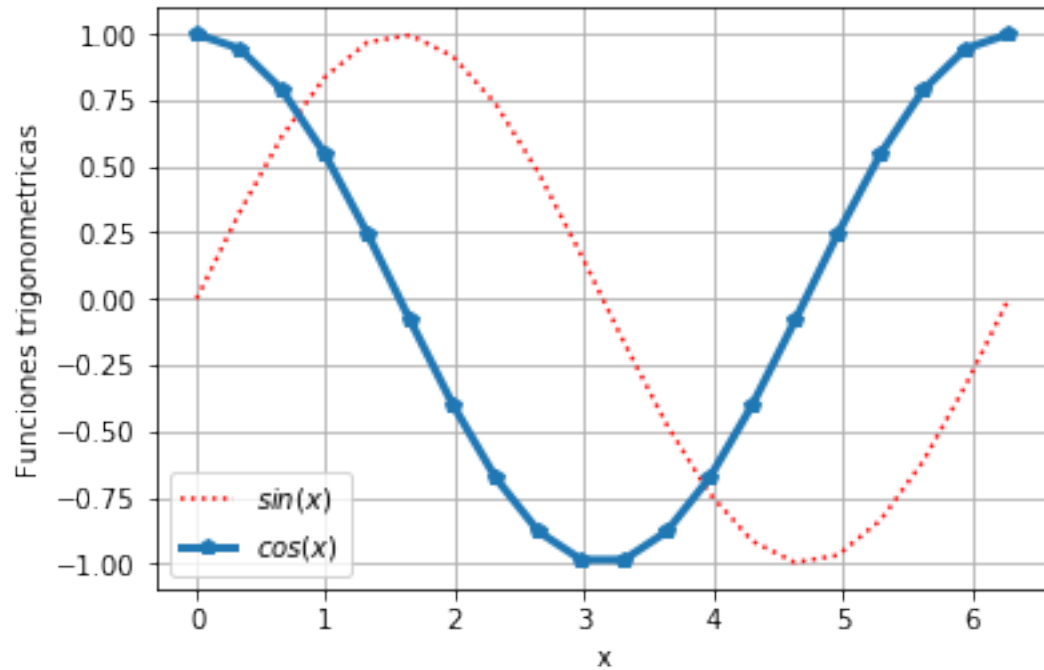
1.8 Gráficar la función senoidal de 0 a 2pi con 15 puntos y un color cafe.

```
In [25]: # Agregamos la libreria de numpy
import numpy as np
# Agregamos la libreria matplotlib
import matplotlib.pyplot as plt
# Generamos el dominio con linspace
x = np.linspace(0,2*np.pi,15)
# Graficamos la funcion
plt.plot(x,np.sin(x),color='g')
# Mostramos en pantalla
plt.show()
```



1.9 Graficar la función senoidal y cosenoidal usando 20 puntos de 0 a 2π y añadiendo todos los parametros vistos hasta ahora.

```
In [3]: # Agregamos la libreria de numpy
import numpy as np
# Agregamos la libreria matplotlib
import matplotlib.pyplot as plt
# Generamos el dominio con linspace
x = np.linspace(0,2*np.pi,20)
# Graficamos la funcion seno
plt.plot(x,np.sin(x),'r:',label='$sin(x)$')
# Graficamos la funcion coseno
plt.plot(x,np.cos(x),marker='p',linewidth=3,label='$cos(x)$')
# Etiqueta del eje x
plt.xlabel('x')
# Etiqueta del eje y
plt.ylabel('Funciones trigonometricas')
# Mostramos la leyenda
plt.legend()
# Mostramos la malla
plt.grid()
# Mostramos en pantalla
plt.show()
```

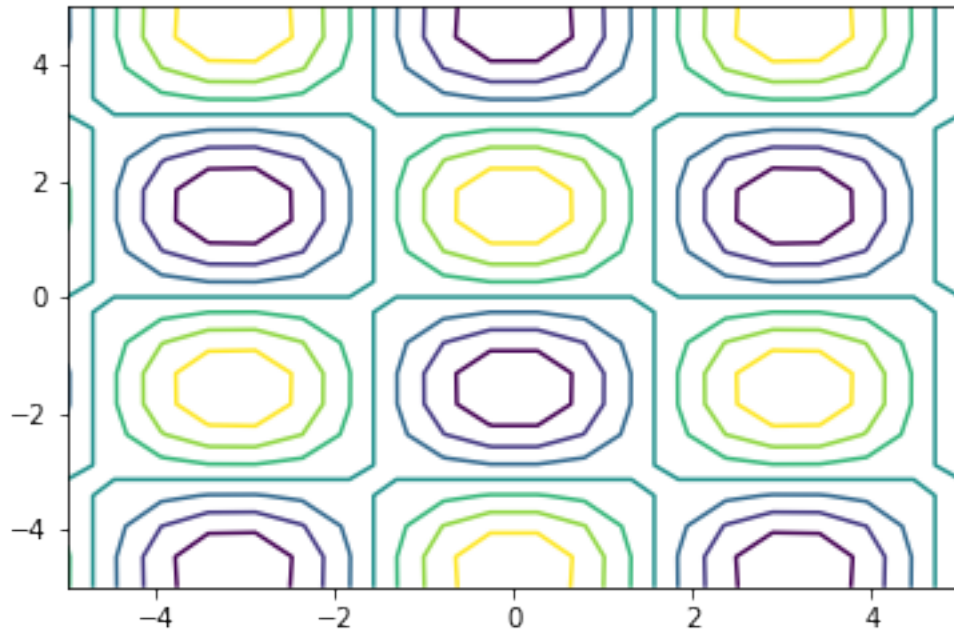


1.10 Contornos

Muestra los ejercicios utilizados para explicar la visualización de curvas de nivel.

1.11 Obtener las curvas de nivel para la gráfica $f(x,y) = \cos(x)\sin(x)$

```
In [51]: # Agregamos las librerías necesarias
import numpy as np
import matplotlib.pyplot as plt
# Generamos el dominio
x = np.linspace(-5,5,20)
y = np.linspace(-5,5,20)
# Generamos la malla
X,Y = np.meshgrid(x,y)
# Graficamos la curva
plt.contour(X,Y,np.cos(X)*np.sin(Y))
# Mostramos el resultado
plt.show()
```



1.12 Ejemplo 2.

```
In [43]: # Agregamos las librerías necesarias
import numpy as np
import matplotlib.pyplot as plt
# Generamos el dominio
x = np.linspace(-5,5,20)
y = np.linspace(-5,5,20)
# Generamos la malla
X,Y = np.meshgrid(x,y)
# Graficamos la curva
plt.contourf(X,Y,np.cos(X)*np.sin(Y))
# Mostramos el resultado
plt.show()
```

