Taller 6

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 27-mar-2020 11:59 PM

[Monica Alejandra Robayo Gonzalez]

[monica.robayo@urosario.edu.co (mailto:monica.robayo@urosario.edu.co)]

Instrucciones:

- Guarde una copia de este *Jupyter Notebook* en su computador, idealmente en una carpeta destinada al material del curso.
- Modifique el nombre del archivo del *notebook*, agregando al final un guión inferior y su nombre y apellido, separados estos últimos por otro guión inferior. Por ejemplo, mi *notebook* se llamaría: mcpp_taller6_santiago_matallana
- Marque el *notebook* con su nombre y e-mail en el bloque verde arriba. Reemplace el texto "[Su nombre acá]" con su nombre y apellido. Similar para su e-mail.
- Desarrolle la totalidad del taller sobre este *notebook*, insertando las celdas que sea necesario debajo de cada pregunta. Haga buen uso de las celdas para código y de las celdas tipo *markdown* según el caso.
- Recuerde salvar periódicamente sus avances.
- Cuando termine el taller:
  1. Descárguelo en PDF. Si tiene algún problema con la conversión, descárguelo en HTML.
  2. Suba todos los archivos a su repositorio en GitHub, en una carpeta destinada exclusivamente para este taller, antes de la fecha y hora límites.

(Todos los ejercicios tienen el mismo valor.)

Resuelva la parte 1 de este documento (http://www.math.pitt.edu/~sussmanm/3040Summer14/exercisesII.pdf).

In [2]:
```
import numpy as np
import scipy.linalg as la
import matplotlib.pyplot as plt
```

1. Choose a value and set the variable x to that value.
2. What is command to compute the square of x? Its cube?

In [2]:
```
x = 20
```

In [4]:
```
# El cuadrado de x
np.power (x, 2)
```

Out[4]:
```
400
```

In [5]:
```
#El cubo de x
np.power (x, 3)
```

Out[5]:
```
8000
```

3. Choose an angle θ and set the variable theta to its value (a number).
4. What is sinθ? cosθ? Angles can be measured in degrees or radians. Which of these are being used used?

In [6]:
```
theta = 180
```

In [7]:
```
np.sin(theta)
```

Out[7]:
```
-0.8011526357338304
```

In [9]:
```
np.cos(theta)
```

Out[9]:
```
-0.5984600690578581
```

R/: Se evidencia que theta se esta midiendo en radianes. Cuando se mide en grados el resultado cambia, dando como resultado -1 para coseno y 0 para seno.

5. Use the np.linspace function to create a row vector called meshPoints containing exactly 500 values with values evenly spaced between -1 and 1.

In [10]:
```
meshPoints = np.linspace(-1,1,500)
print (meshPoints)
```

```
[-1.         -0.99599198 -0.99198397 -0.98797595 -0.98396794 -0.97995992
 -0.9759519  -0.97194389 -0.96793587 -0.96392786 -0.95991984 -0.95591182
 -0.95190381 -0.94789579 -0.94388778 -0.93987976 -0.93587174 -0.93186373
 -0.92785571 -0.9238477  -0.91983968 -0.91583166 -0.91182365 -0.90781563
 -0.90380762 -0.8997996  -0.89579158 -0.89178357 -0.88777555 -0.88376754
 -0.87975952 -0.8757515  -0.87174349 -0.86773547 -0.86372745 -0.85971944
 -0.85571142 -0.85170341 -0.84769539 -0.84368737 -0.83967936 -0.83567134
 -0.83166333 -0.82765531 -0.82364729 -0.81963928 -0.81563126 -0.81162325
 -0.80761523 -0.80360721 -0.7995992  -0.79559118 -0.79158317 -0.78757515
 -0.78356713 -0.77955912 -0.7755511  -0.77154309 -0.76753507 -0.76352705
 -0.75951904 -0.75551102 -0.75150301 -0.74749499 -0.74348697 -0.73947896
 -0.73547094 -0.73146293 -0.72745491 -0.72344689 -0.71943888 -0.71543086
 -0.71142285 -0.70741483 -0.70340681 -0.6993988  -0.69539078 -0.69138277
 -0.68737475 -0.68336673 -0.67935872 -0.6753507  -0.67134269 -0.66733467
 -0.66332665 -0.65931864 -0.65531062 -0.65130261 -0.64729459 -0.64328657
 -0.63927856 -0.63527054 -0.63126253 -0.62725451 -0.62324649 -0.61923848
 -0.61523046 -0.61122244 -0.60721443 -0.60320641 -0.5991984  -0.59519038
 -0.59118236 -0.58717435 -0.58316633 -0.57915832 -0.5751503  -0.57114228
 -0.56713427 -0.56312625 -0.55911824 -0.55511022 -0.5511022  -0.54709419
 -0.54308617 -0.53907816 -0.53507014 -0.53106212 -0.52705411 -0.52304609
 -0.51903808 -0.51503006 -0.51102204 -0.50701403 -0.50300601 -0.498998
 -0.49498998 -0.49098196 -0.48697395 -0.48296593 -0.47895792 -0.4749499
 -0.47094188 -0.46693387 -0.46292585 -0.45891784 -0.45490982 -0.4509018
 -0.44689379 -0.44288577 -0.43887776 -0.43486974 -0.43086172 -0.42685371
 -0.42284569 -0.41883768 -0.41482966 -0.41082164 -0.40681363 -0.40280561
 -0.3987976  -0.39478958 -0.39078156 -0.38677355 -0.38276553 -0.37875752
 -0.3747495  -0.37074148 -0.36673347 -0.36272545 -0.35871743 -0.35470942
 -0.3507014  -0.34669339 -0.34268537 -0.33867735 -0.33466934 -0.33066132
 -0.32665331 -0.32264529 -0.31863727 -0.31462926 -0.31062124 -0.30661323
 -0.30260521 -0.29859719 -0.29458918 -0.29058116 -0.28657315 -0.28256513
 -0.27855711 -0.2745491  -0.27054108 -0.26653307 -0.26252505 -0.25851703
 -0.25450902 -0.250501   -0.24649299 -0.24248497 -0.23847695 -0.23446894
 -0.23046092 -0.22645291 -0.22244489 -0.21843687 -0.21442886 -0.21042084
 -0.20641283 -0.20240481 -0.19839679 -0.19438878 -0.19038076 -0.18637275
 -0.18236473 -0.17835671 -0.1743487  -0.17034068 -0.16633267 -0.16232465
 -0.15831663 -0.15430862 -0.1503006  -0.14629259 -0.14228457 -0.13827655
 -0.13426854 -0.13026052 -0.12625251 -0.12224449 -0.11823647 -0.11422846
 -0.11022044 -0.10621242 -0.10220441 -0.09819639 -0.09418838 -0.09018036
 -0.08617234 -0.08216433 -0.07815631 -0.0741483  -0.07014028 -0.06613226
 -0.06212425 -0.05811623 -0.05410822 -0.0501002  -0.04609218 -0.04208417
 -0.03807615 -0.03406814 -0.03006012 -0.0260521  -0.02204409 -0.01803607
 -0.01402806 -0.01002004 -0.00601202 -0.00200401  0.00200401  0.00601202
  0.01002004  0.01402806  0.01803607  0.02204409  0.0260521   0.03006012
  0.03406814  0.03807615  0.04208417  0.04609218  0.0501002   0.05410822
  0.05811623  0.06212425  0.06613226  0.07014028  0.0741483   0.07815631
  0.08216433  0.08617234  0.09018036  0.09418838  0.09819639  0.10220441
  0.10621242  0.11022044  0.11422846  0.11823647  0.12224449  0.12625251
  0.13026052  0.13426854  0.13827655  0.14228457  0.14629259  0.1503006
  0.15430862  0.15831663  0.16232465  0.16633267  0.17034068  0.1743487
  0.17835671  0.18236473  0.18637275  0.19038076  0.19438878  0.19839679
  0.20240481  0.20641283  0.21042084  0.21442886  0.21843687  0.22244489
  0.22645291  0.23046092  0.23446894  0.23847695  0.24248497  0.24649299
  0.250501    0.25450902  0.25851703  0.26252505  0.26653307  0.27054108
  0.2745491   0.27855711  0.28256513  0.28657315  0.29058116  0.29458918
```

```
0.29859719  0.30260521  0.30661323  0.31062124  0.31462926  0.31863727
0.32264529  0.32665331  0.33066132  0.33466934  0.33867735  0.34268537
0.34669339  0.3507014   0.35470942  0.35871743  0.36272545  0.36673347
0.37074148  0.3747495   0.37875752  0.38276553  0.38677355  0.39078156
0.39478958  0.3987976   0.40280561  0.40681363  0.41082164  0.41482966
0.41883768  0.42284569  0.42685371  0.43086172  0.43486974  0.43887776
0.44288577  0.44689379  0.4509018   0.45490982  0.45891784  0.46292585
0.46693387  0.47094188  0.4749499   0.47895792  0.48296593  0.48697395
0.49098196  0.49498998  0.498998    0.50300601  0.50701403  0.51102204
0.51503006  0.51903808  0.52304609  0.52705411  0.53106212  0.53507014
0.53907816  0.54308617  0.54709419  0.5511022   0.55511022  0.55911824
0.56312625  0.56713427  0.57114228  0.5751503   0.57915832  0.58316633
0.58717435  0.59118236  0.59519038  0.5991984   0.60320641  0.60721443
0.61122244  0.61523046  0.61923848  0.62324649  0.62725451  0.63126253
0.63527054  0.63927856  0.64328657  0.64729459  0.65130261  0.65531062
0.65931864  0.66332665  0.66733467  0.67134269  0.6753507   0.67935872
0.68336673  0.68737475  0.69138277  0.69539078  0.6993988   0.70340681
0.70741483  0.71142285  0.71543086  0.71943888  0.72344689  0.72745491
0.73146293  0.73547094  0.73947896  0.74348697  0.74749499  0.75150301
0.75551102  0.75951904  0.76352705  0.76753507  0.77154309  0.7755511
0.77955912  0.78356713  0.78757515  0.79158317  0.79559118  0.7995992
0.80360721  0.80761523  0.81162325  0.81563126  0.81963928  0.82364729
0.82765531  0.83166333  0.83567134  0.83967936  0.84368737  0.84769539
0.85170341  0.85571142  0.85971944  0.86372745  0.86773547  0.87174349
0.8757515   0.87975952  0.88376754  0.88777555  0.89178357  0.89579158
0.8997996   0.90380762  0.90781563  0.91182365  0.91583166  0.91983968
0.9238477   0.92785571  0.93186373  0.93587174  0.93987976  0.94388778
0.94789579  0.95190381  0.95591182  0.95991984  0.96392786  0.96793587
0.97194389  0.9759519   0.97995992  0.98396794  0.98797595  0.99198397
0.99599198  1.        ]
```

6. What expression will yield the value of the 53th element of meshPoints? What is this value?

In [12]:
```
print(meshPoints[52])
```

```
-0.7915831663326653
```

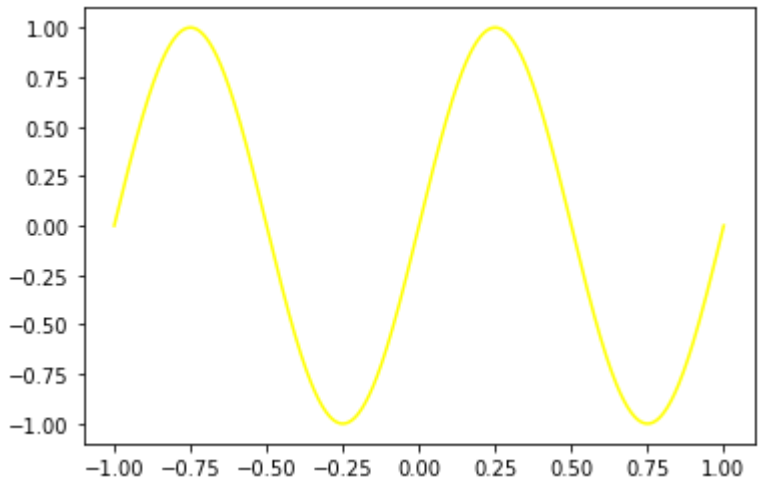
7. Produce a plot of a sinusoid on the interval [−1,1] using the command plt.plot(meshPoints,np.sin(2*pi*meshPoints)) Please save this plot as a jpeg (.jpg) file and send it along with your work.

In [16]:
```
import math
```

In [19]:
```
math.pi
```

Out[19]:
```
3.141592653589793
```

In [33]:
```
plt.plot(meshPoints,np.sin(2*math.pi*meshPoints), color = "yellow" ) ;
```



In [35]:
```
plt.savefig("seno.jpg")
```

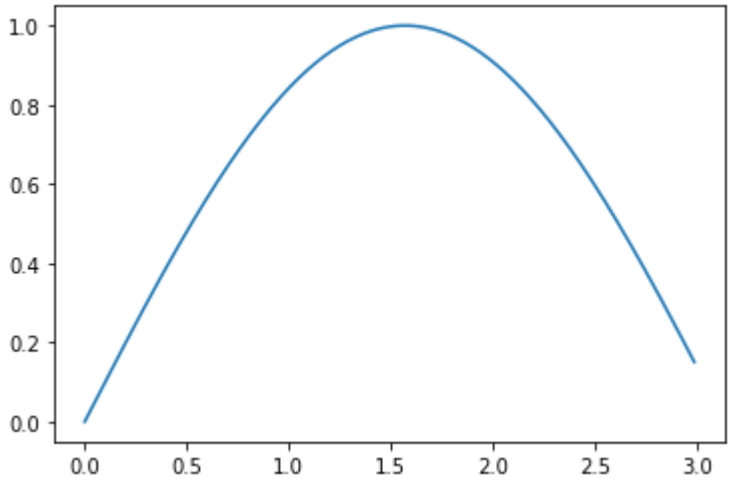
```
<Figure size 432x288 with 0 Axes>
```

Resuelva los ejercicios de las secciones 4.1, 5.1, 6.1, 7.4 y 8.5 de este documento (http://www.python-academy.com/download/pycon2012/matplotlib_handout.pdf).

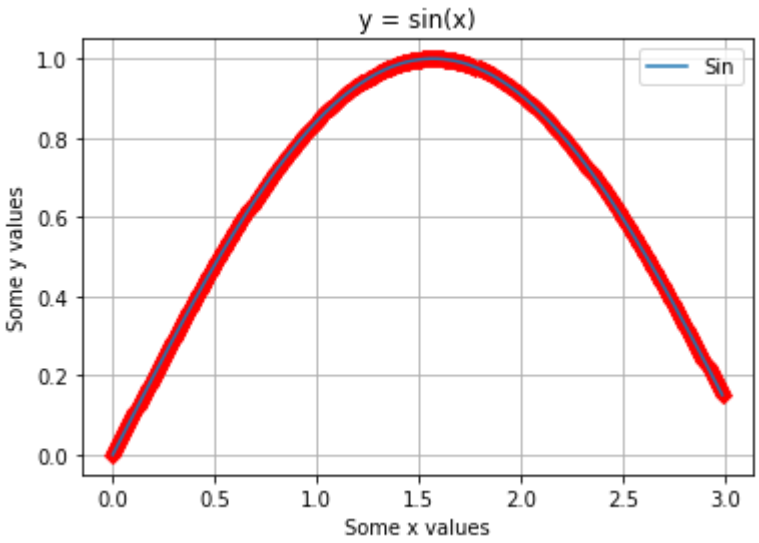
4.1 Exercises

1. Plot a simple graph of a sinus function in the range 0 to 3 with a step size of 0.01.
2. Make the line red. Add diamond-shaped markers with size of 5.
3. Add a legend and a grid to the plot.

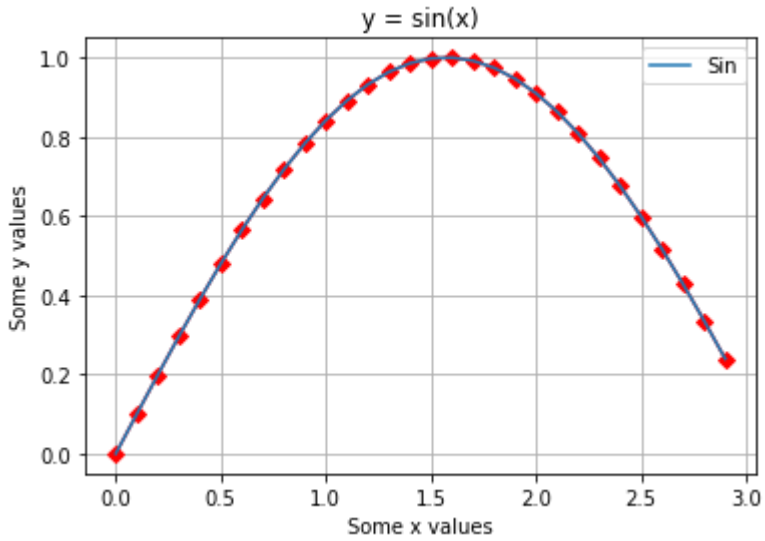
In [48]:
```
x = np.arange(0, 3, 0.01);
y = np.sin(x)
plt.plot(x,y) ;
```



In [123]:
```
x = np.arange(0, 3, 0.01)
y = np.sin(x)
plt.title("y = sin(x)")
plt.xlabel("Some x values")
plt.ylabel("Some y values")
plt.plot(x, y, color="r", marker="D", markersize="5")
plt.plot(x,y, label = "Sin")
plt.grid()
plt.legend() ;
```



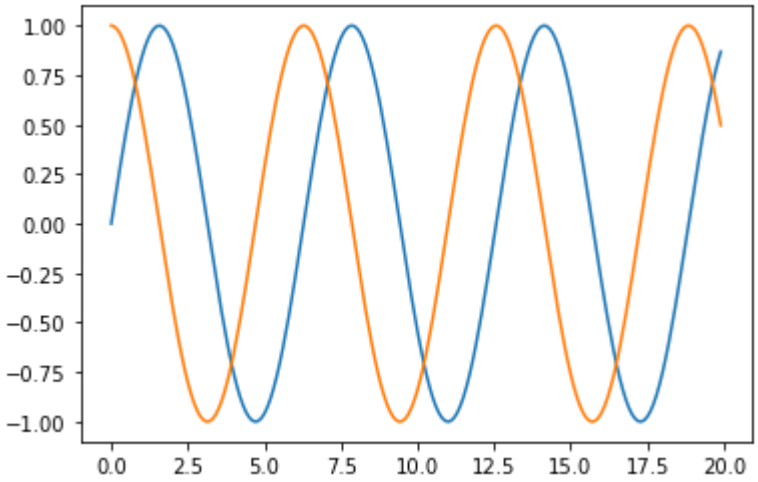
In [124]:
```
#otra forma para visualizar mejor el diamond-shaped markers es cambiando el step

x = np.arange(0, 3, 0.1)
y = np.sin(x)
plt.title("y = sin(x)")
plt.xlabel("Some x values")
plt.ylabel("Some y values")
plt.plot(x, y, color="r", marker="D", markersize="5")
plt.plot(x,y, label = "Sin")
plt.grid()
plt.legend() ;
```



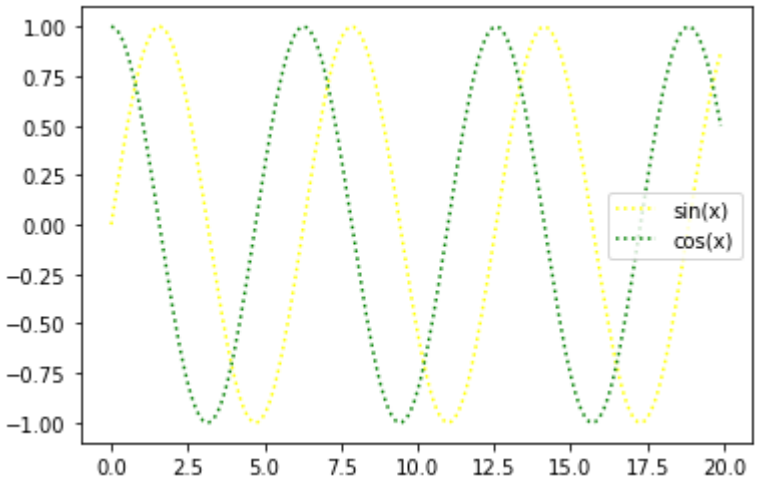
5.1 Exercise

1. Apply different line styles to a plot. Change line color and thickness as well as the size and the kind of the marker. Experiment with different styles.

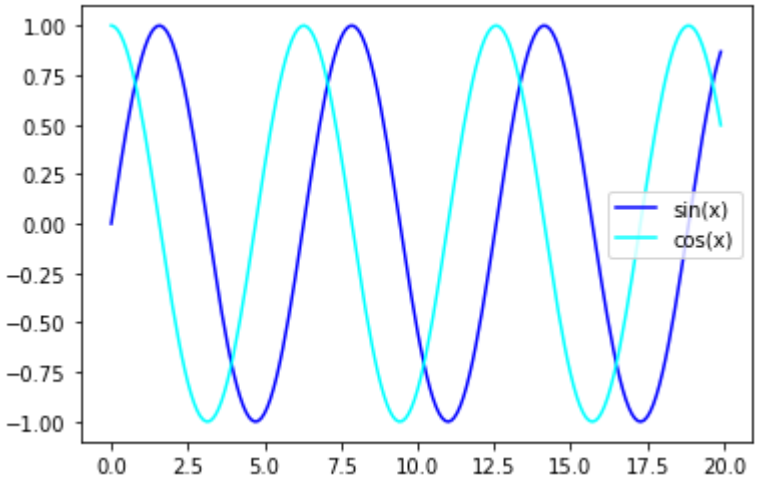
In [140]:
```
x = np.arange(0, 20, 0.1);
plt.plot(x, np.sin(x))
plt.plot(x, np.cos(x)) ;
```



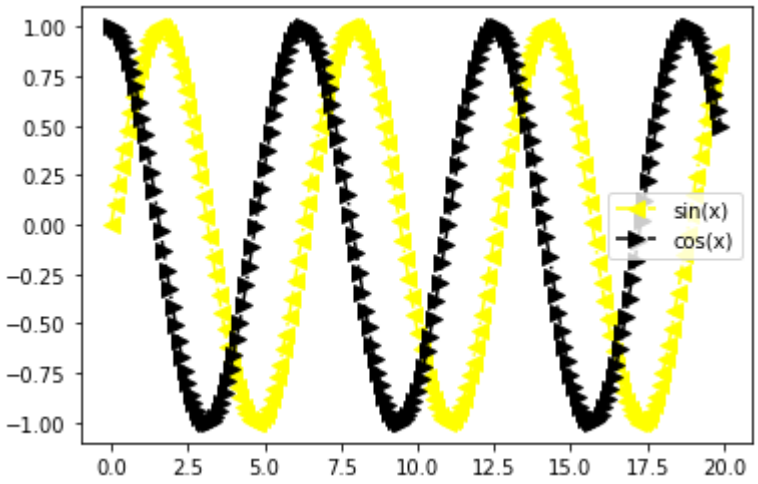
In [145]:
```
plt.plot(x, np.sin(x), color="yellow", label="sin(x)", linestyle=":", markersize=
plt.plot(x, np.cos(x), color="green", label="cos(x)", linestyle=":", markersize=!
plt.legend();
```



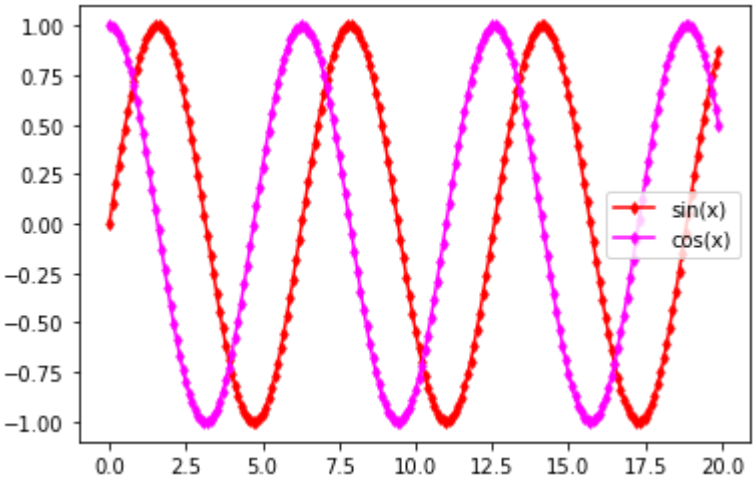
In [147]:
```
plt.plot(x, np.sin(x), color="blue", label="sin(x)", linestyle="-", markersize=3
plt.plot(x, np.cos(x), color="cyan", label="cos(x)", linestyle="-", markersize=3
plt.legend();
```



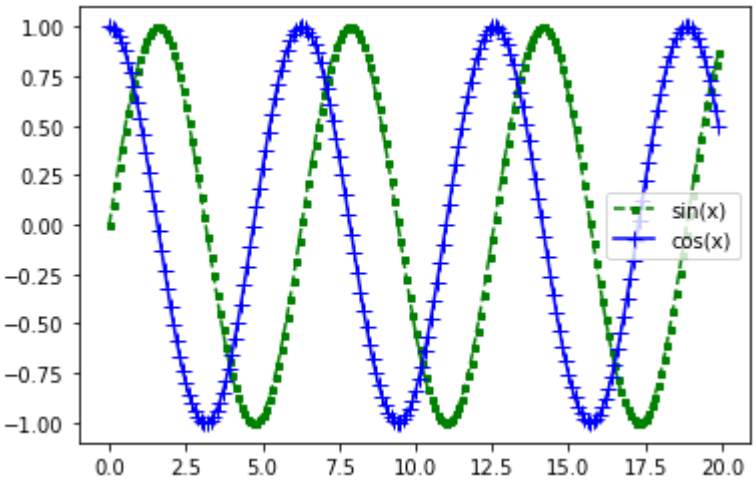
In [154]:
```
plt.plot(x, np.sin(x), color="yellow", label="sin(x)", linestyle="-.", markersize
plt.plot(x, np.cos(x), color="black", label="cos(x)", linestyle="-.", markersize=
plt.legend();
```



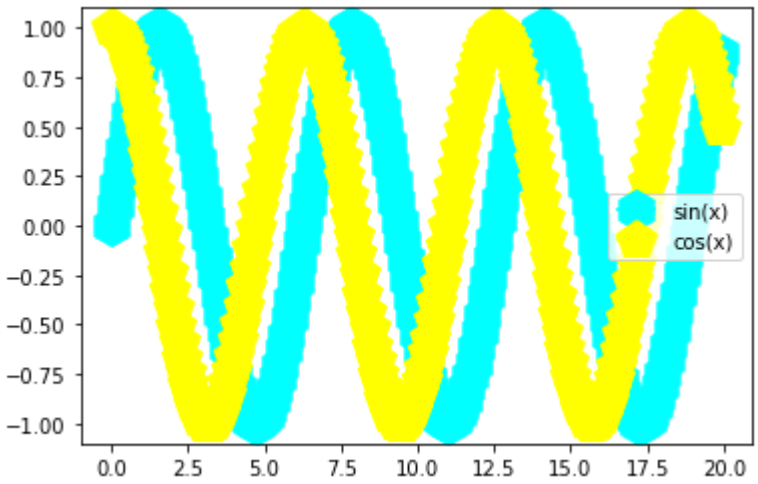
In [155]:
```
plt.plot(x, np.sin(x), color="red", label="sin(x)", linestyle="-", markersize=4,
plt.plot(x, np.cos(x), color="magenta", label="cos(x)", linestyle="-", markersiz
plt.legend();
```



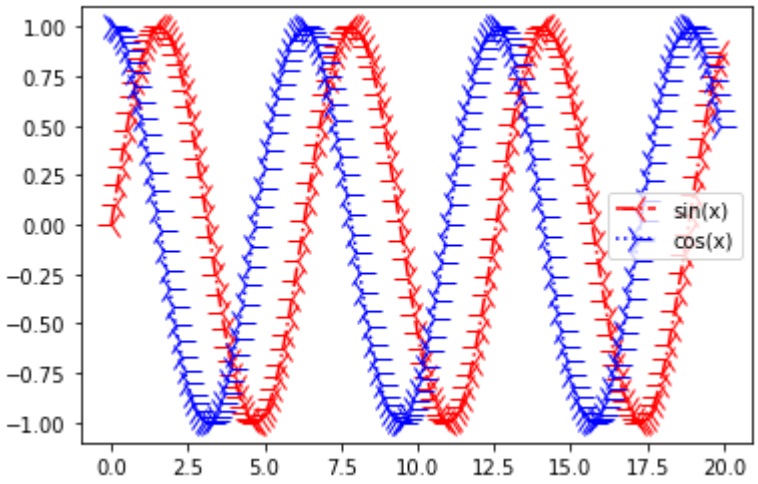
In [157]:
```
plt.plot(x, np.sin(x), color="green", label="sin(x)", linestyle="--", markersize
plt.plot(x, np.cos(x), color="blue", label="cos(x)", linestyle="-", markersize=8
plt.legend();
```



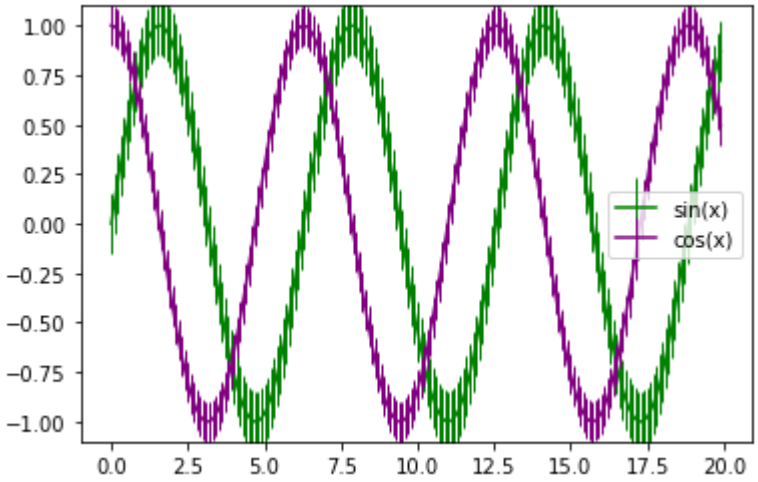
In [162]:
```
plt.plot(x, np.sin(x), color="cyan", label="sin(x)", linestyle="-.", markersize=
plt.plot(x, np.cos(x), color="yellow", label="cos(x)", linestyle=":", markersize=
plt.legend();
```



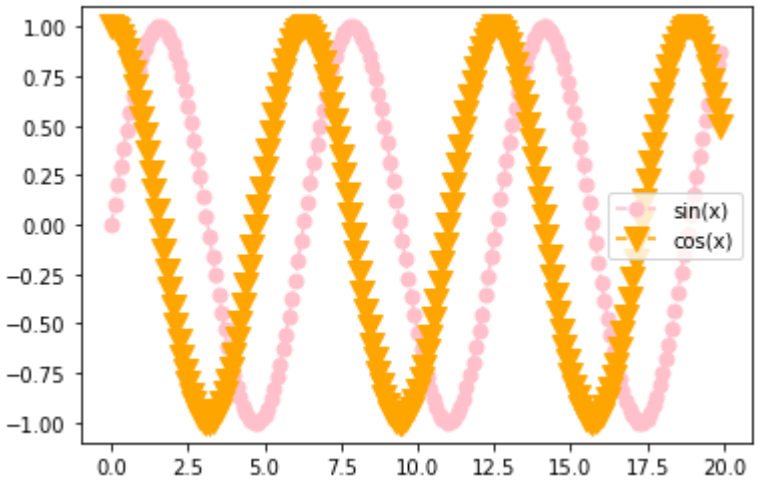
In [167]:
```
plt.plot(x, np.sin(x), color="red", label="sin(x)", linestyle="-.", markersize=1
plt.plot(x, np.cos(x), color="blue", label="cos(x)", linestyle=":", markersize=1
plt.legend();
```



In [174]:
```
plt.plot(x, np.sin(x), color="green", label="sin(x)", linestyle="-", markersize=
plt.plot(x, np.cos(x), color="purple", label="cos(x)", linestyle="-", markersize
plt.legend();
```



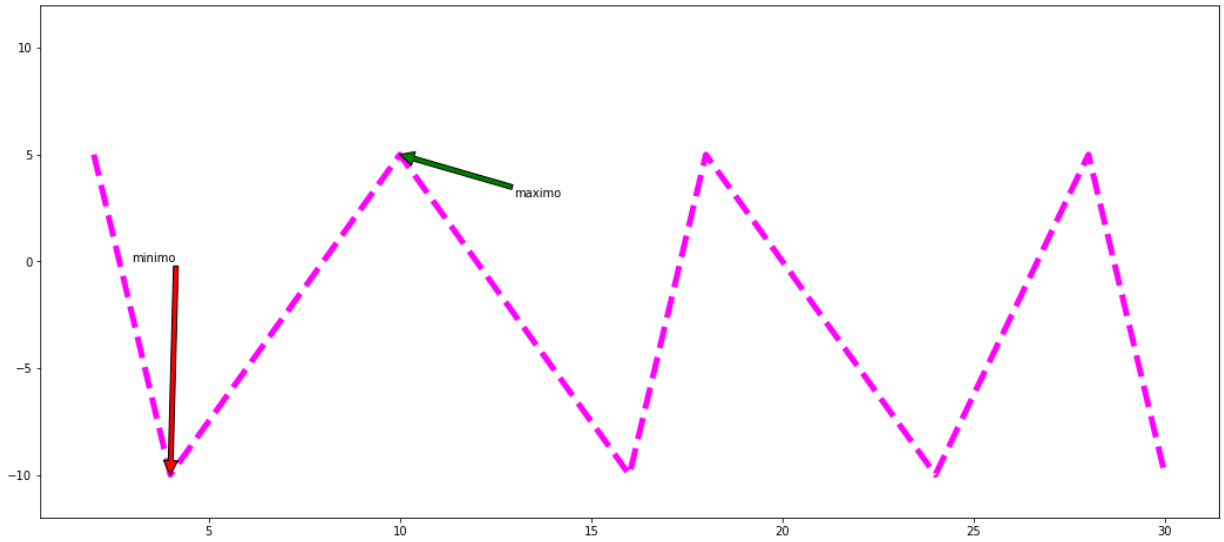
In [184]:
```
plt.plot(x, np.sin(x), color="pink", label="sin(x)", linestyle="--", markersize=
plt.plot(x, np.cos(x), color="orange", label="cos(x)", linestyle="--", markersiz
plt.legend();
```



6.1 Exercise

1. Annotate a line at two places with text. Use green and red arrows and align it according to figure points and data.

In [73]:
```
fig, ax = plt.subplots()

x = [2,4,10,16,18,24,28,30]
y = [5,-10,5,-10,5,-10,5,-10]
line, = ax.plot(x, y, linestyle="--", color= "magenta", lw=5)

ax.annotate ("minimo", xy = (4, -10), xytext=(3,0), arrowprops={'facecolor': 'r'
ax.annotate ("maximo", xy = (10, 5), xytext=(13,3), arrowprops={'facecolor': 'g'

ax.set_ylim(-12, 12)
plt.show()
```

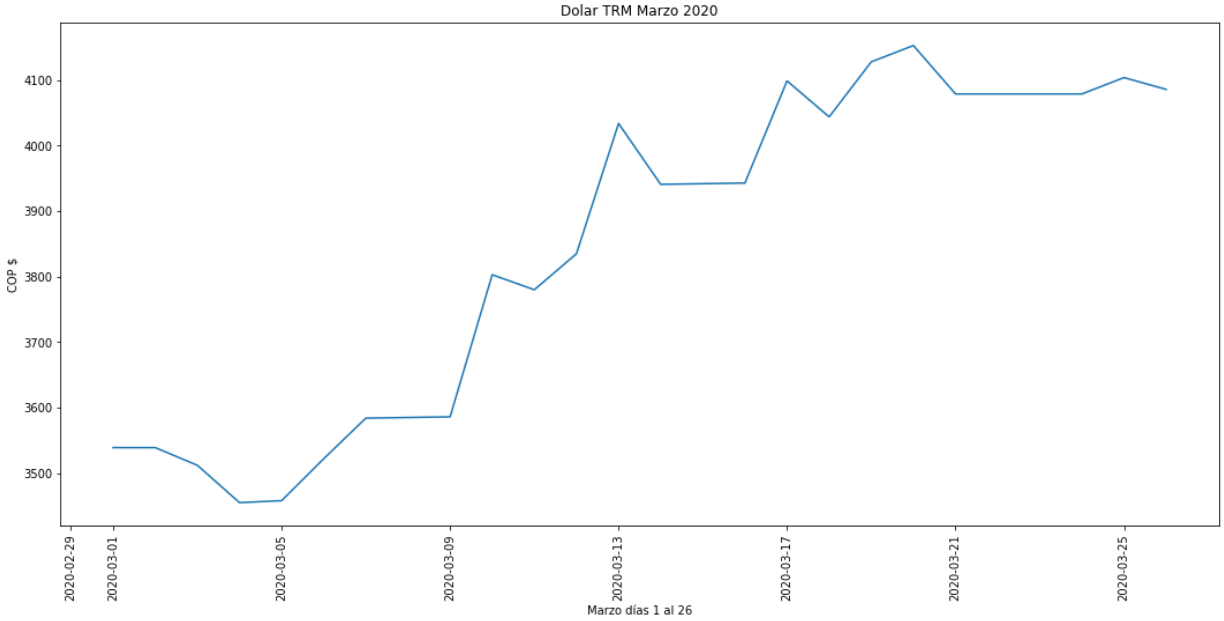


7.4 Exercises

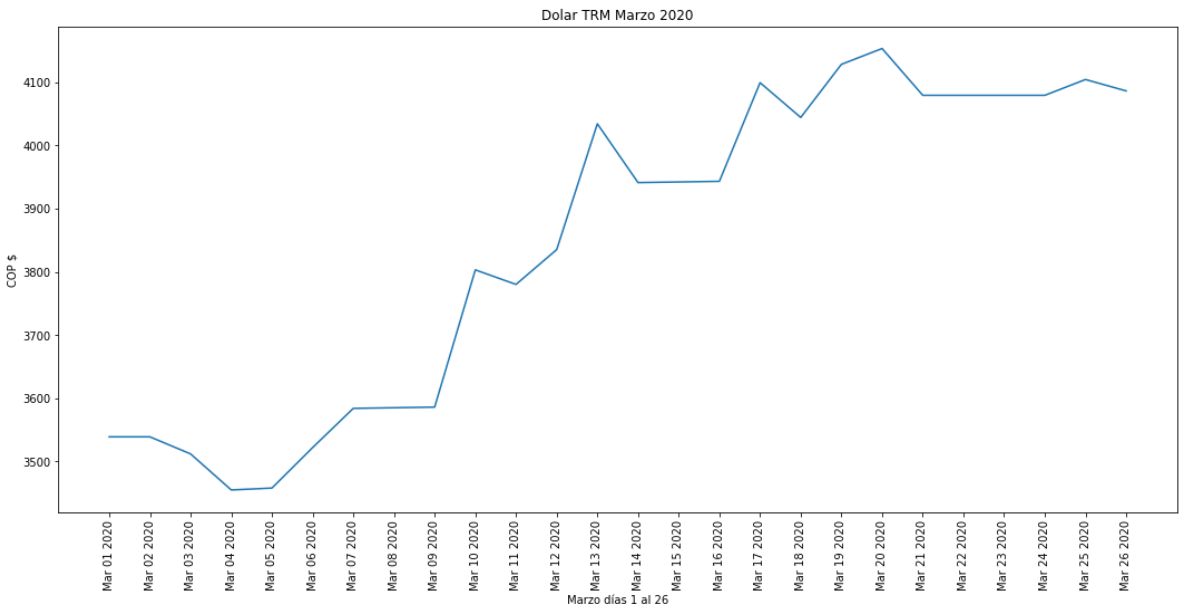
1. Plot a graph with dates for one year with daily values at the x axis using the built-in module datetime.
2. Format the dates in such a way that only the first day of the month is shown.
3. Display the dates with and without the year. Show the month as number and as first three letters of the month name.

1. Plot a graph with dates for one year with daily values at the x axis using the built-in module datetime.

In [53]:
```
#1 punto: una forma de hacerlo es usando la siguiente estructura datetime.date(ye
import datetime

plt.title("Dolar TRM Marzo 2020")

plt.xlabel("Marzo días 1 al 26")
plt.ylabel("COP $")

trm = [3539,3539,3512,3455,3458, 3522,3584,3585,3586,3803,3780,3835, 4034, 3941,
date = [datetime.date(2020,3, i) for i in range (1,27)]

plt.xticks(rotation=90)

plt.plot(date, trm) ;
```

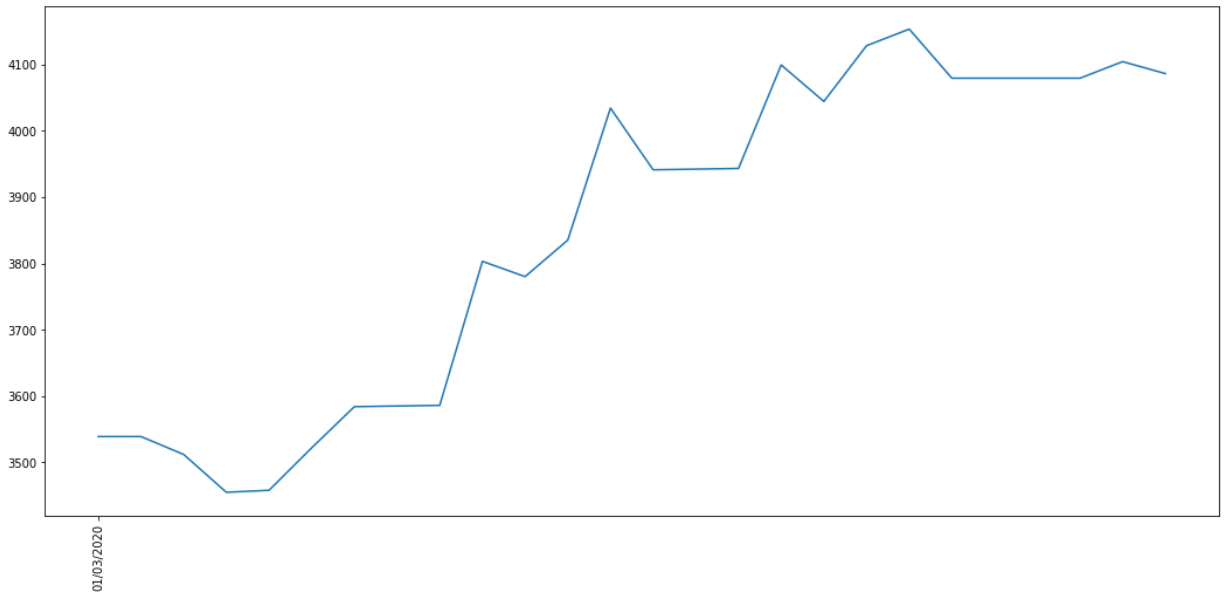


In [48]:
```
# otra forma de hacerlo es usar la estructura datetime.datetime(year,month,day) p
import datetime

plt.title("Dolar TRM Marzo 2020")

trm = [3539,3539,3512,3455,3458, 3522,3584,3585,3586,3803,3780,3835, 4034, 3941,3

date = [datetime.datetime(2020, 3, 1).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 2).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 3).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 4).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 5).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 6).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 7).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 8).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 9).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 10).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 11).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 12).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 13).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 14).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 15).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 16).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 17).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 18).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 19).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 20).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 21).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 22).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 23).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 24).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 25).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 26).strftime("%b %d %Y")]


plt.xlabel("Marzo días 1 al 26")
plt.ylabel("COP $")

plt.xticks(rotation=90)

plt.plot(date, trm) ;
```

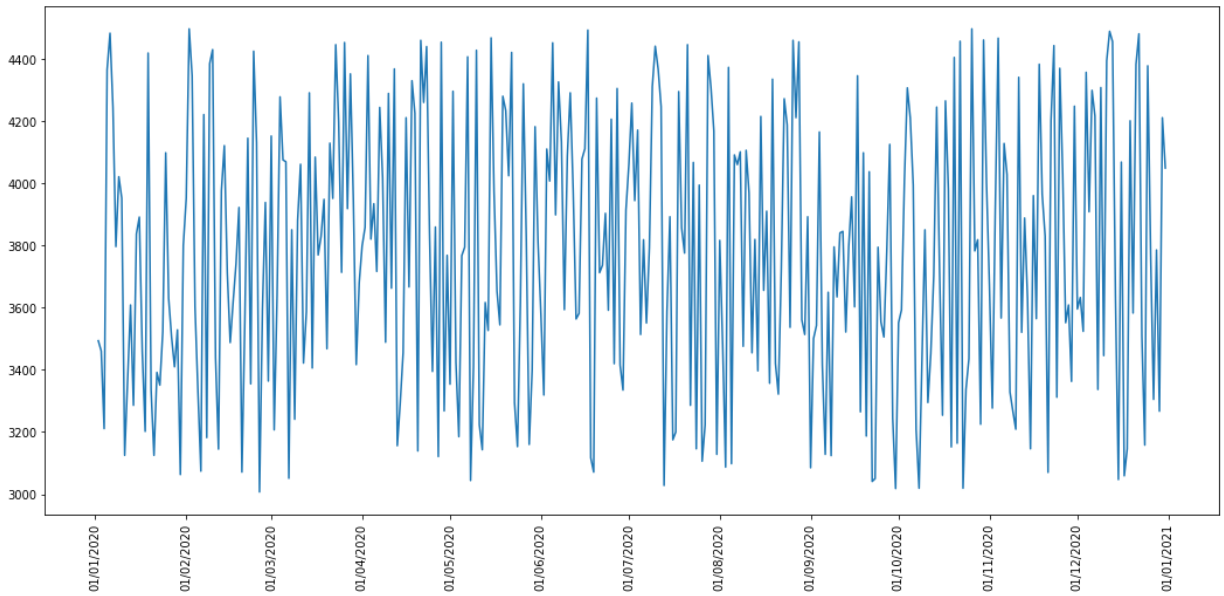


2. Format the dates in such a way that only the first day of the month is shown.

In [69]:
```
#una forma de hacerlo mostrando solo un mes
from matplotlib.ticker import (MultipleLocator, FormatStrFormatter,
                               AutoMinorLocator, NullFormatter)

from matplotlib.dates import (DateFormatter, DayLocator, MonthLocator)

fig, ax = plt.subplots()

ax.xaxis.set_major_locator(MonthLocator())
ax.xaxis.set_major_formatter(DateFormatter('%d/%m/%Y'))

plt.plot(date, trm) ;
plt.xticks(rotation=90)
plt.show()
```

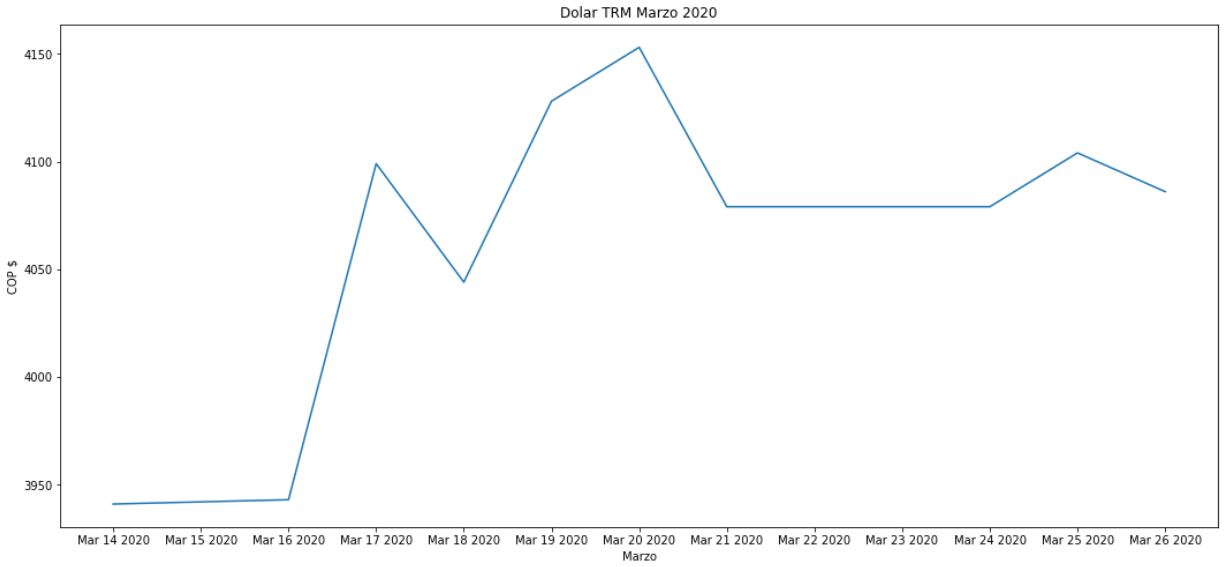


In [74]:
```
#otra forma de hacerlo es usando un random randint para todo el año 2020

from matplotlib.ticker import (MultipleLocator, FormatStrFormatter,
                               AutoMinorLocator, NullFormatter)

from matplotlib.dates import (DateFormatter, DayLocator, MonthLocator)

import random

fig, ax = plt.subplots()

ax.xaxis.set_major_locator(MonthLocator())
ax.xaxis.set_major_formatter(DateFormatter('%d/%m/%Y'))

trm = [ random.randint(3000, 4500) for i in range (1,366) ]
ini = datetime.date(2020,1,1)
date = [ini + datetime.timedelta(days=i) for i in range (1,366)]

plt.plot(date, trm) ;
plt.xticks(rotation=90)
plt.show()
```

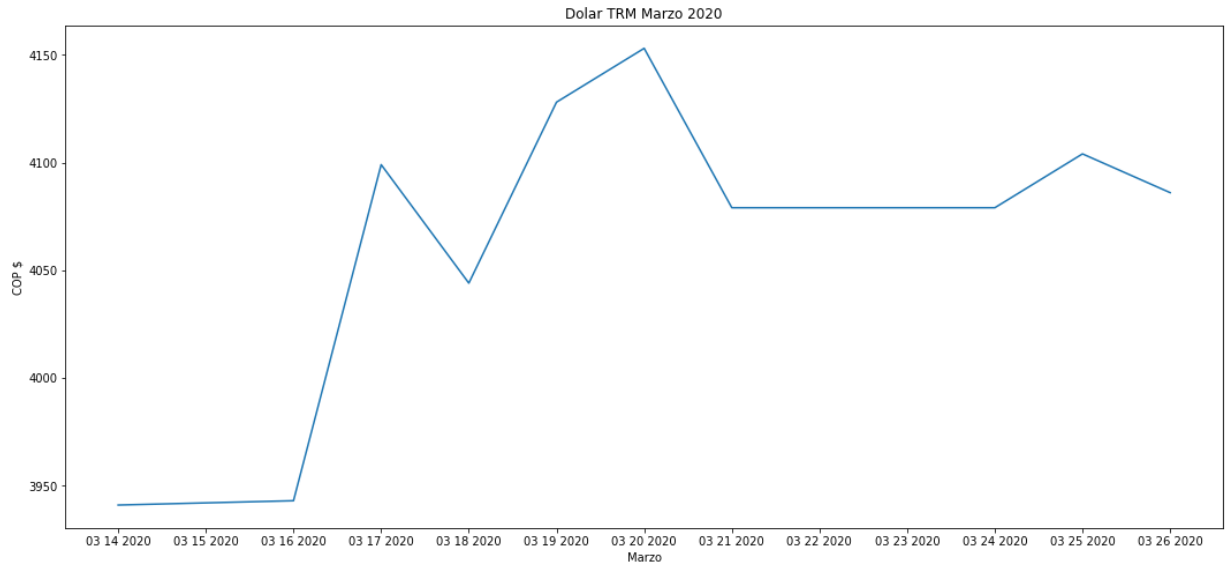


3. Display the dates with and without the year. Show the month as number and as first three letters of the month name.

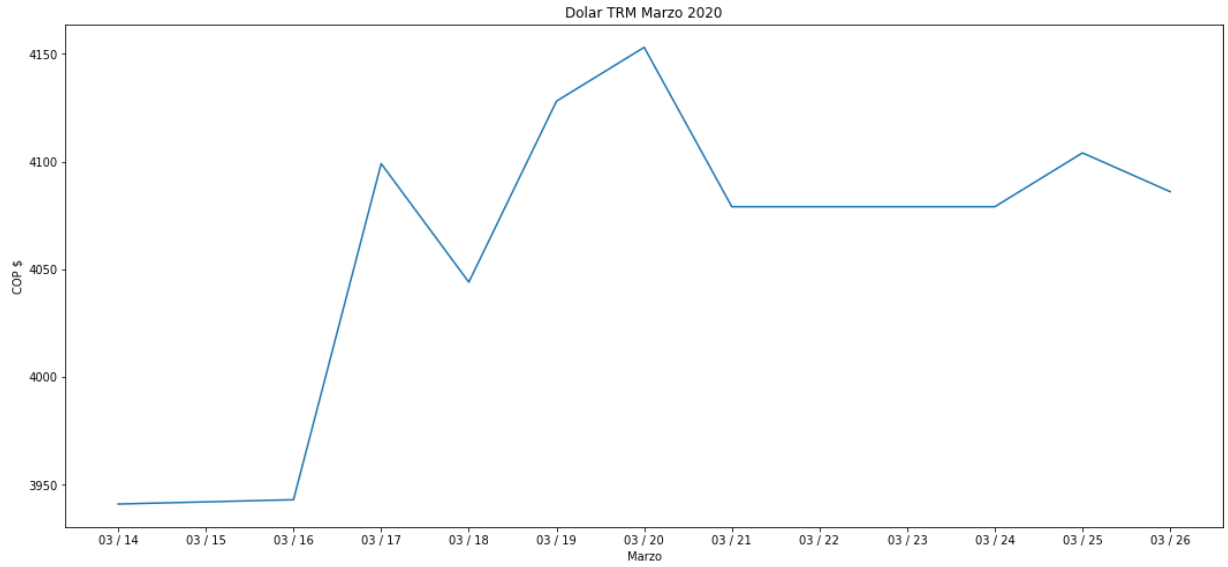
In [71]:
```
# mes con las primeras tres letras y con año y día
import datetime

plt.title("Dolar TRM Marzo 2020")

trm = [3941,3942,3943,4099, 4044, 4128, 4153, 4079,4079,4079,4079,4104,4086]

date = [datetime.datetime(2020, 3, 14).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 15).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 16).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 17).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 18).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 19).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 20).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 21).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 22).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 23).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 24).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 25).strftime("%b %d %Y"),
  datetime.datetime(2020, 3, 26).strftime("%b %d %Y")]

plt.xlabel("Marzo")
plt.ylabel("COP $")

plt.plot(date, trm) ;
```



In [277]:
```
# mes como numero
import datetime

plt.title("Dolar TRM Marzo 2020")

trm = [3941,3942,3943,4099, 4044, 4128, 4153, 4079,4079,4079,4079,4104,4086]

date = [datetime.datetime(2020, 3, 14).strftime("%m %d %Y"),
  datetime.datetime(2020, 3, 15).strftime("%m %d %Y"),
  datetime.datetime(2020, 3, 16).strftime("%m %d %Y"),
  datetime.datetime(2020, 3, 17).strftime("%m %d %Y"),
  datetime.datetime(2020, 3, 18).strftime("%m %d %Y"),
  datetime.datetime(2020, 3, 19).strftime("%m %d %Y"),
  datetime.datetime(2020, 3, 20).strftime("%m %d %Y"),
  datetime.datetime(2020, 3, 21).strftime("%m %d %Y"),
  datetime.datetime(2020, 3, 22).strftime("%m %d %Y"),
  datetime.datetime(2020, 3, 23).strftime("%m %d %Y"),
  datetime.datetime(2020, 3, 24).strftime("%m %d %Y"),
  datetime.datetime(2020, 3, 25).strftime("%m %d %Y"),
  datetime.datetime(2020, 3, 26).strftime("%m %d %Y")]

plt.xlabel("Marzo")
plt.ylabel("COP $")

plt.plot(date, trm) ;
```



In [281]:
```
# fecha sin año
import datetime

plt.title("Dolar TRM Marzo 2020")

trm = [3941,3942,3943,4099, 4044, 4128, 4153, 4079,4079,4079,4079,4104,4086]

date = [datetime.datetime(2020, 3, 14).strftime("%m / %d"),
  datetime.datetime(2020, 3, 15).strftime("%m / %d"),
  datetime.datetime(2020, 3, 16).strftime("%m / %d"),
  datetime.datetime(2020, 3, 17).strftime("%m / %d"),
  datetime.datetime(2020, 3, 18).strftime("%m / %d"),
  datetime.datetime(2020, 3, 19).strftime("%m / %d"),
  datetime.datetime(2020, 3, 20).strftime("%m / %d"),
  datetime.datetime(2020, 3, 21).strftime("%m / %d"),
  datetime.datetime(2020, 3, 22).strftime("%m / %d"),
  datetime.datetime(2020, 3, 23).strftime("%m / %d"),
  datetime.datetime(2020, 3, 24).strftime("%m / %d"),
  datetime.datetime(2020, 3, 25).strftime("%m / %d"),
  datetime.datetime(2020, 3, 26).strftime("%m / %d")]

plt.xlabel("Marzo")
plt.ylabel("COP $")

plt.plot(date, trm) ;
```

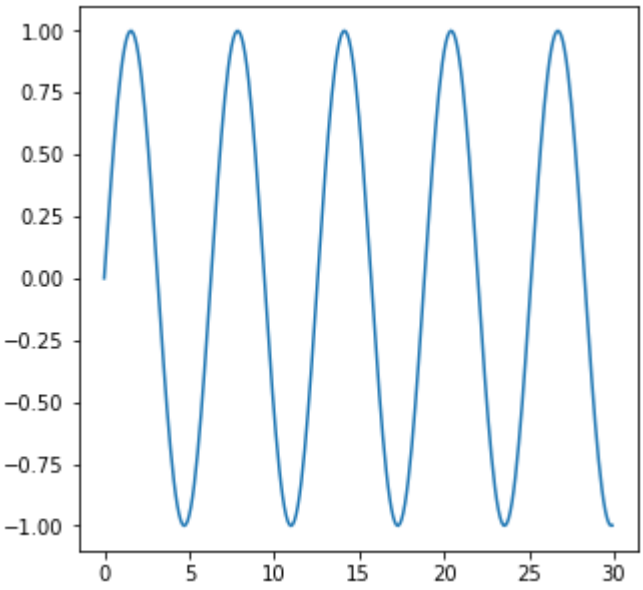


8.5 Exercises

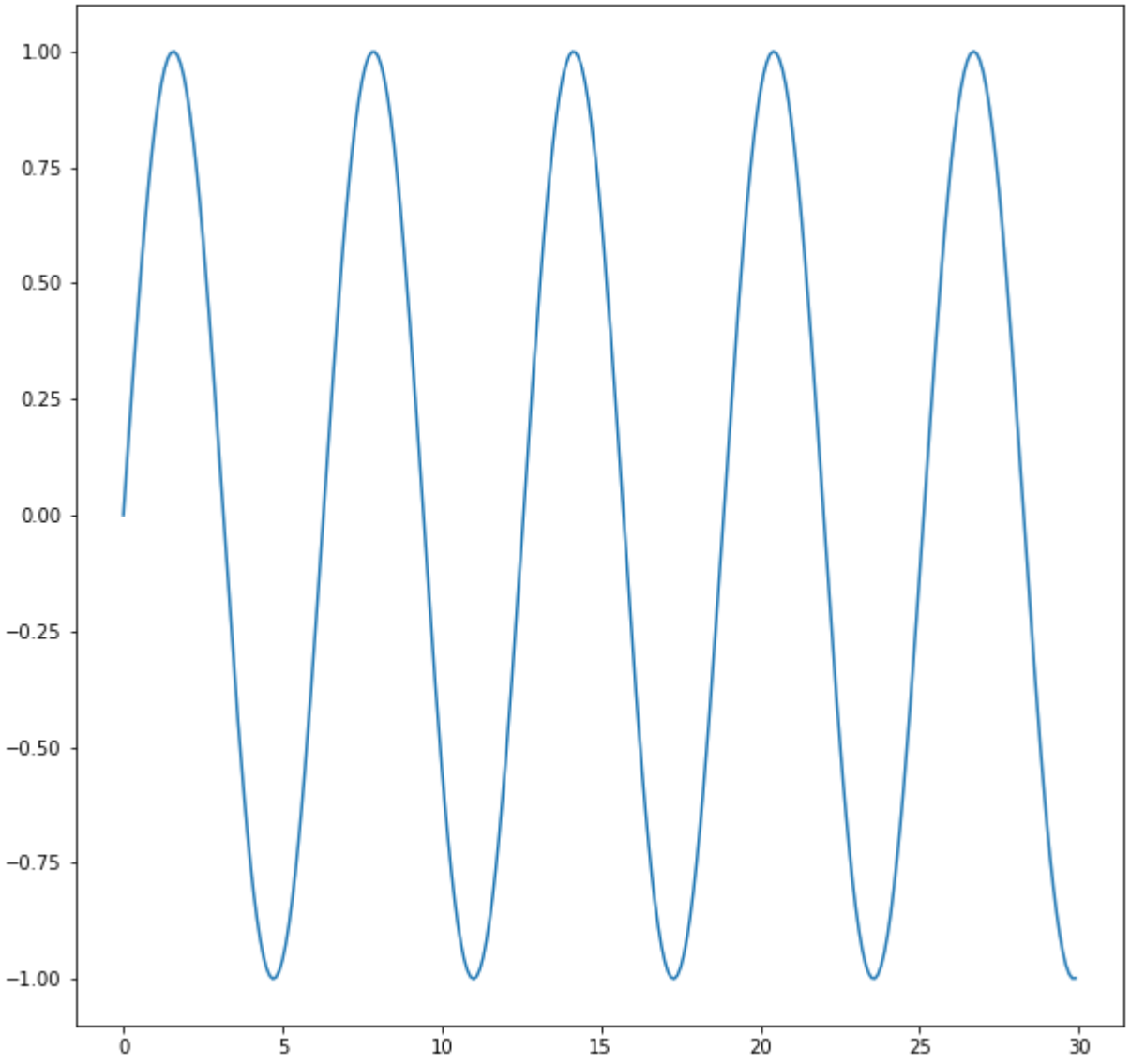
1. Draw two figures, one 5 by 5, one 10 by 10 inches.
2. Add four subplots to one figure. Add labels and ticks only to the outermost axes.
3. Place a small plot in one bigger plot

1. Draw two figures, one 5 by 5, one 10 by 10 inches.

In [11]:
```
x = np.arange(0, 30, 0.1);

fig= plt.figure(figsize=(5,5))
plt.plot(x, np.sin(x));
```

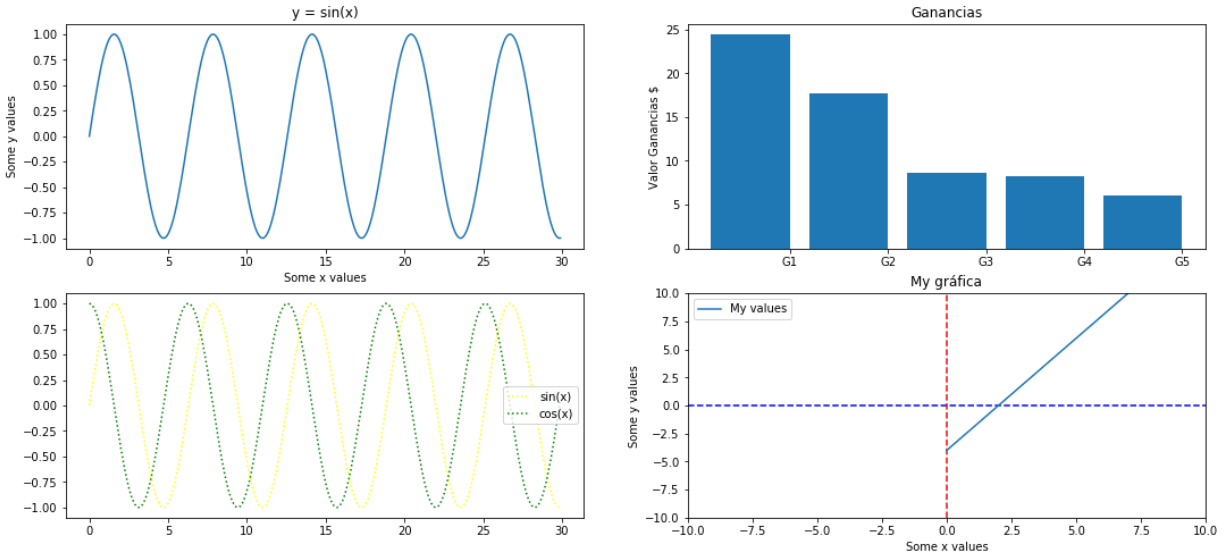


In [10]:
```
x = np.arange(0, 30, 0.1);

fig= plt.figure(figsize=(10,10))
plt.plot(x, np.sin(x));
```



2. Add four subplots to one figure. Add labels and ticks only to the outermost axes.

In [17]:
```
plt.rcParams["figure.figsize"] = [18.0, 8.0]

plt.subplot(221)
x = np.arange(0, 30, 0.1)
plt.title("y = sin(x)")
plt.xlabel("Some x values")
plt.ylabel("Some y values")
plt.plot(x, np.sin(x))

plt.subplot(222)
x = ["G1", "G2", "G3", "G4", "G5"]
values = [24.4, 17.7, 8.6, 8.3, 6.1]
plt.title("Ganancias")
plt.ylabel("Valor Ganancias $")
xs = np.arange(len(x))
xtick_locs = xs + 0.4
plt.xticks(xtick_locs, x)
plt.bar(xs, values);

plt.subplot(223)
x = np.arange(0, 30, 0.1);
plt.plot(x, np.sin(x), color="yellow", label="sin(x)", linestyle=":")
plt.plot(x, np.cos(x), color="green", label="cos(x)", linestyle=":")
plt.legend();

plt.subplot(224)
xs = [-4,-2, 0, 2, 4, 6, 8, 10]
plt.title("My gráfica")
plt.xlabel("Some x values")
plt.ylabel("Some y values")
plt.axhline(0, color="blue", linestyle="--")
plt.axvline(0, color="red", linestyle="--")
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.plot(xs, label = "My values")
plt.legend();
```



3. Place a small plot in one bigger plot

In [41]:
```
fig = plt.figure()
x = np.arange(0, 10, 0.1);
y = np.arange(0, 15, 0.1);

axes1 = fig.add_axes([0.1, 0.1, 0.9, 0.9]) # main axes
axes2 = fig.add_axes([0.2, 0.5, 0.25, 0.25]) # inset axes

# main figure
axes1.plot(x, np.sin(x), 'y')
axes1.set_xlabel('some values x')
axes1.set_ylabel('some values y')
axes1.set_title("Sin")

# insert
axes2.plot(y, np.cos(y), 'g')
axes2.set_xlabel('some values x')
axes2.set_ylabel('some valyes y')
axes2.set_title('Cos');
```