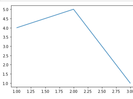
Python matplotlib basic tutorial

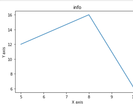
Generating Graph

In [1]:
```
# here some basic code to generate one of the most simple graph
from matplotlib import pyplot as plt

# alerting to our canvas
plt.plot([1,2,3],[4,5,1])

# showing what we plioted
plt.show()
```



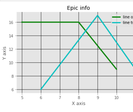
Title and Lebels

In [2]:
```
# Let's add title and labels to our graph
from matplotlib import pyplot as plt

x=[5,8,10]
y=[12,16,6]

plt.plot(x,y)

plt.title('info')
plt.ylabel('Y axis')
plt.xlabel('X axis')

plt.show()
```



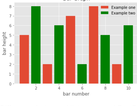
Adding style to our graph

In [3]:
```
# Adding style to our graph
from matplotlib import pyplot as plt
from matplotlib import style

style.use('ggplot')

x=[5,8,10]
y=[16,16,9]

x2=[6,9,11]
y2=[6,15,7]

plt.plot(x,y,'g',label='line one', linewidth=3)
plt.plot(x2,y2,'c',label='line two', linewidth=3)

plt.title('Epic info')
plt.ylabel('Y axis')
plt.xlabel('X axis')

plt.legend()

plt.grid(True, color='k')

plt.show()
```



Bar Graphs

In [4]:
```
# Bar Graphs
# So Bar graphs are basically used to compare things between different groups, and are trying to measure changes over
# time bar graphs are very well suit when changes are larger.
import matplotlib.pyplot as plt

plt.bar([1,3,5,7,9],[5,2,7,8,2],label="Example one")
plt.bar([2,4,6,8,10],[8,6,2,5,6],label="Example two",color='g')

plt.legend()
plt.xlabel("bar number")
plt.ylabel("bar height")

plt.title('Bar Graph')

plt.show()
```

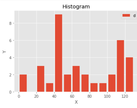


Difference between Histogram and Bar Graph

In histogram we have quantitative variables and when i talking about bar graphs they have categorical variables.
Let explain with some examples
e.g So if suppose i went a plot, the GDP growth of every city in a particular country, so at that time I'll use a bar plot because it has a categories this particular city like new jerzy, new york allthose things.

Now when i talk about i'll use histogram when i am talking about quantitative variables that mean i am talking about age group. In the some example if i wana calculate how much each age group is actually contributing toward GDP growth that time i'll be using histogram.

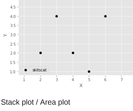
In [5]:
```
# Histogram
import matplotlib.pyplot as plt

population_ages=[22,55,62,45,21,22,34,42,42,4,8,99,102,115,110,120,117,121,
                 132,122,130,111,105,116,112,85,75,65,54,44,42,43,42,45,46,67,77]

bins=[0,10,20,30,40,50,60,70,80,90,100,110,120,130]

plt.hist(population_ages, bins,histtype='bar', rwidth=0.8)

plt.xlabel('X')
plt.ylabel('Y')
plt.title('Histogram')
plt.legend('data') #means legend me jo bhi dalenge hum wo side me dikhega jo ki aap delh sakte hai chart me orange color me d.
plt.show()
```



Scatter plot

Now before we understand how to plot scatter graph we need to understand why we actually use scatter plot.
Usually we are use scatter plot an order to compare two variables if you plotting in three dimension looking for a co-relation or graphs.
So basically you tried to fiend out how much two or three variables related to each other.

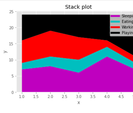
In [6]:
```
# Scatter plot
import matplotlib.pyplot as plt

x=[1,2,3,4,5,6,7,8]
y=[5,2,4,2,1,4,5,2]

plt.scatter(x,y,label='skitscat',color='k')

plt.xlabel('X')
plt.ylabel('Y')
plt.title('scatter plot')
plt.legend()
plt.show()
```



Stack plot / Area plot

So basically these are graphs are very similar to the line graph, they can be used to track changes over time for one or more graphs.
Area graphs are good to use when you are tracking the changes in two or more relative graphs that make uo one whole categories.

In [7]:
```
# Stack plot/ Area plot
import matplotlib.pyplot as plt

days=[1,2,3,4,5]

sleeping=[7,8,6,11,7]
eating=[2,3,4,3,2]
working=[7,8,7,2,2]
playing=[8,5,7,8,13]

plt.plot([],[], color='m', label='Sleeping',linewidth=5)
plt.plot([],[], color='c', label='Eating',linewidth=5)
plt.plot([],[], color='r', label='Working',linewidth=5)
plt.plot([],[], color='k', label='Playing',linewidth=5)

plt.stackplot(days, sleeping,eating,working,playing, colors=['m', 'c', 'r', 'k'])

plt.xlabel('x')
plt.ylabel('y')
plt.title('Stack plot')
plt.legend()
plt.show()
```



Pie Chart

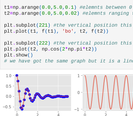
Pie chart are pretty much similar to stack plots. Only they are for a certain point in time. Typically a pie chart is used to show paths to the whole and often a pecentage share. You can consider the examples of percentage of marketshare and things like that.

In [8]:
```
# Pie chart
import matplotlib.pyplot as plt

slices = [7,2,2,13]
activities = ['sleeping', 'eating', 'working', 'playing']
cols = ['c', 'm', 'r', 'b']

plt.pie(slices, #relevant sizes for each part
        labels = activities,
        colors= cols, #for the corresponding slices
        startangle = 90, #That means first division will be vertical line
        shadow = True,
        explode = (0,0.1,0,0), #for pull out slices
        autopct = '%1.1f%%')
plt.title('Pie plot')
plt.show()
```

Working with multiple plots

subplot: it has subs to plot multiple plots. So when we write subplot(211) that means we have two plots. Horizontally we have only one plot present and vertically we have two plots.

In [9]:
```
import numpy as np
import matplotlib.pyplot as plt

def f(t):
    return np.exp(-t) * np.cos(2*np.pi*t)

t1=np.arange(0.0,5.0,0.1) #elements between 0 till 5 and the step of 0.1.
t2=np.arange(0.0,5.0,0.02) #elements ranging from 0 till 5 and its step of 0.02

plt.subplot(211) #horizontally we have only one plot present and vertically we have two plots. And the vertical position this plot will be our firs
plt.plot(t1, f(t1), 'bo', t2, f(t2))

plt.subplot(212) #horizontally we have only one plot present and vertically we have two plots. And the vertical position this plot will be our seco
plt.plot(t2, np.cos(2*np.pi*t2))
plt.show()
```



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

def f(t):
    return np.exp(-t) * np.cos(2*np.pi*t)

t1=np.arange(0.0,5.0,0.1) #elements between 0 till 5 and the step of 0.1.
t2=np.arange(0.0,5.0,0.02) #elements ranging from 0 till 5 and its step of 0.02

plt.subplot(221) #the vertical position this plot will be our first graph. "differences are there"
plt.plot(t1, f(t1), 'bo', t2, f(t2))

plt.subplot(222) #the vertical position this plot will be our second graph. "differences are there"
plt.plot(t2, np.cos(2*np.pi*t2))
plt.show()
# we have got the same graph but it is a line horizontally
```


In [ ]:
```

```