Numpy :

Numpy is a low level library written in c (and fortran) for high level mathematical functions Numpy cleverly overcomes the problems of running slower algorithms on python by using multidimensionsal arrays and functions that operate on arrays Any algorithms can then be expressed as a function on arrays allowing the algorithms to be rum quickly.

N-d array properties.

##A powerful N_dimensional array object. ##sophisticated (broadcasting) functions. ##tools for intergrating C/C++ and Fortran code. ##useful linear algebra , Fouries transform and random number capabilities,

application of numpy.

1. Homogeneous .
2. Only be number (Number , float , complex) 3.Fixed item size

In [1]:
```
# creating Numpy Arrays.
#1. Usinf np.array() //1-D and 2-D
#2.Using np.zeros /ones/empty/random()
#3.usingnp.arange()
#4.Usingnp.linspace()
#5.Using copy()
#6. Using identity()
```

In [3]:
```
# creating numpy/n-d arrays
import numpy as np
```

In [3]:
```
arr1 = np.array([1,2,3,4,5,6])
```

In [4]:
```
arr1
```

Out[4]:
```
array([1, 2, 3, 4, 5, 6])
```

In [5]:
```
type(arr1)
```

Out[5]:
```
numpy.ndarray
```

In [6]:
```
arr2 = np.array([[1,2,3],[4,5,6]])
```

In [7]:
```
arr2
```

Out[7]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```

In [8]:
```
arr3 = np.zeros([2,3])
```

In [9]:
```
arr3
```

Out[9]:
```
array([[0., 0., 0.],
       [0., 0., 0.]])
```

In [10]:
```
arr4 = np.ones([3,3])
```

In [11]:
```
arr4
```

Out[11]:
```
array([[1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.]])
```

In [12]:
```
arr5 = np.identity(5)
```

In [13]:
```
arr5
```

Out[13]:
```
array([[1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1.]])
```

In [14]:
```
arr6 = np.arange(10)
```

In [15]:
```
arr6
```

Out[15]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [16]:
```
arrx = np.arange(16)
arrx
```

Out[16]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15])
```

In [17]:
```
arry = np.arange(5,16,2)
arry
```

Out[17]:
```
array([ 5,  7,  9, 11, 13, 15])
```

In [18]:
```
arr7 = np.linspace(10,20,10)
arr7
```

Out[18]:
```
array([10.        , 11.11111111, 12.22222222, 13.33333333, 14.44444444,
       15.55555556, 16.66666667, 17.77777778, 18.88888889, 20.        ])
```

In [19]:
```
arr8 = arr7.copy()
arr8
```

Out[19]:
```
array([10.        , 11.11111111, 12.22222222, 13.33333333, 14.44444444,
       15.55555556, 16.66666667, 17.77777778, 18.88888889, 20.        ])
```

In [20]:
```
# properties & Attributes
#1.Shape
#2.nDim
#3.Size
#4.Itemsize
#5.Dtype
#6.astype()
```

In [21]:
```
arr1.shape
```

Out[21]:
```
(6,)
```

In [22]:
```
arr2.shape
```

Out[22]:
```
(2, 3)
```

In [23]:
```
arr9 = np.array([[1,2],[3,4],
                 [5,6],[7,8]])
arr9
```

Out[23]:
```
array([[1, 2],
       [3, 4],
       [5, 6],
       [7, 8]])
```

In [24]:
```
arr9.shape
```

Out[24]:
```
(4, 2)
```

In [4]:
```
arr10= np.array([[[1,2],[3,4]],[[5,6],[7,8]]])
```

In [26]:
```
arr6.ndim
```

Out[26]:
```
1
```

In [27]:
```
arr1.ndim
```

Out[27]:
```
1
```

In [28]:
```
arr9.shape
```

Out[28]:
```
(4, 2)
```

In [29]:
```
arr9.itemsize
```

Out[29]:
```
4
```

In [30]:
```
arr9.size
```

Out[30]:
```
8
```

In [31]:
```
arr9.dtype
```

Out[31]:
```
dtype('int32')
```

In [32]:
```
arr9.astype(float)
```

Out[32]:
```
array([[1., 2.],
       [3., 4.],
       [5., 6.],
       [7., 8.]])
```

In [33]:
```
# Numpy vs Python array
```

In [34]:
```
# Lists vs Arrays
#1.Faster
#2.Convenient
#3.Less Memory
```

In [35]:
```
lista= range(100)
arr11=np.arange(100)
```

In [36]:
```
import sys
```

In [37]:
```
print(sys.getsizeof(87)*len(lista))
```

```
2800
```

In [38]:
```
print(arr11.itemsize*arr11.size)
```

```
400
```

In [39]:
```
import time
```

In [40]:
```
x= range(100000)
y=range(100000,200000)

start_time=time_time()

c=[(x,y)forx,y in zip(x,y) ]
print(time.time()-start_time) #part 5 time 7.24
print(time.time()-start_time)
```

```
  Cell In[40], line 6
    c=[(x,y)forx,y in zip(x,y) ]
       ^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
```

In [41]:
```
a = np.arange(10000000)
b = np.arange(10000000,20000000)

start_time=time.time()
c = a+b

print(time.time()-start_time)
```

```
0.05995631217956543
```

In [42]:
```
# indexing slicing and iteration
```

In [43]:
```
arr12=np.arange(24)
arr12
```

Out[43]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23])
```

In [44]:
```
arr12=np.arange(24).reshape(12,2)
arr12
```

Out[44]:
```
array([[ 0,  1],
       [ 2,  3],
       [ 4,  5],
       [ 6,  7],
       [ 8,  9],
       [10, 11],
       [12, 13],
       [14, 15],
       [16, 17],
       [18, 19],
       [20, 21],
       [22, 23]])
```

In [45]:
```
arr12=np.arange(24).reshape(6,4)
arr12
```

Out[45]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15],
       [16, 17, 18, 19],
       [20, 21, 22, 23]])
```

In [46]:
```
arr12=np.arange(24).reshape(8,3)
arr12
```

Out[46]:
```
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11],
       [12, 13, 14],
       [15, 16, 17],
       [18, 19, 20],
       [21, 22, 23]])
```

In [47]:
```
arr1
```

Out[47]:
```
array([1, 2, 3, 4, 5, 6])
```

In [48]:
```
arr1[2:4]
```

Out[48]:
```
array([3, 4])
```

In [49]:
```
arr12[2]
```

Out[49]:
```
array([6, 7, 8])
```

In [50]:
```
arr12[: 2]
```

Out[50]:
```
array([[0, 1, 2],
       [3, 4, 5]])
```

In [51]:
```
arr12[:,2]
```

Out[51]:
```
array([ 2,  5,  8, 11, 14, 17, 20, 23])
```

In [52]:
```
arr12=np.arange(24).reshape(6,4)
```

In [53]:
```
arr12
```

Out[53]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15],
       [16, 17, 18, 19],
       [20, 21, 22, 23]])
```

In [54]:
```
arr12[2:4,1:3]
```

Out[54]:
```
array([[ 9, 10],
       [13, 14]])
```

In [55]:
```
# iteration
```

In [56]:
```
arr12
```

Out[56]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15],
       [16, 17, 18, 19],
       [20, 21, 22, 23]])
```

In [57]:
```
for i in arr12:
    print(i)
```

```
[0 1 2 3]
[4 5 6 7]
[ 8  9 10 11]
[12 13 14 15]
[16 17 18 19]
[20 21 22 23]
```

In [58]:
```
for i in np.nditer(arr12):
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
```

In [59]:
```
# Part 7
```

In [60]:
```
import numpy as np
```

In [61]:
```
#operation on array
```

In [62]:
```
arr1 = np.array([1,2,3,4,5,6])
arr2 = np.array([4,5,6,7,8,9])
```

In [63]:
```
arr1-arr2
```

Out[63]:
```
array([-3, -3, -3, -3, -3, -3])
```

In [64]:
```
arr1*arr2  # vector multiplication
```

Out[64]:
```
array([ 4, 10, 18, 28, 40, 54])
```

In [65]:
```
arr1*2 #scelar multiplication
```

Out[65]:
```
array([ 2,  4,  6,  8, 10, 12])
```

In [66]:
```
arr2>3
```

Out[66]:
```
array([ True,  True,  True,  True,  True,  True])
```

In [67]:
```
arr3=np.arange(6).reshape(2,3)
arr4=np.arange(6,12).reshape(3,2)
```

In [68]:
```
arr3.dot(arr4)
```

Out[68]:
```
array([[ 28,  31],
       [100, 112]])
```

In [69]:
```
arr1.dot(arr2) # dot product
```

Out[69]:
```
154
```

In [70]:
```
arr1.dot(arr3)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[70], line 1
----> 1 arr1.dot(arr3)

ValueError: shapes (6,) and (2,3) not aligned: 6 (dim 0) != 2 (dim 0)
```

In [72]:
```
arr4
```

Out[72]:
```
array([[ 6,  7],
       [ 8,  9],
       [10, 11]])
```

In [73]:
```
arr4.max()
```

Out[73]:
```
11
```

In [74]:
```
arr4.min()
```

Out[74]:
```
6
```

In [75]:
```
arr4.min(axis=0)
```

Out[75]:
```
array([6, 7])
```

In [ ]:
```

```

Type *Markdown* and LaTeX: α^2

In [76]:
```
arr4.min(axis=1)
```

Out[76]:
```
array([ 6,  8, 10])
```

In [77]:
```
arr4.sum(axis=0)
```

Out[77]:
```
array([24, 27])
```

In [78]:
```
arr4.mean()
```

Out[78]:
```
8.5
```

In [79]:
```
arr4.std()
```

Out[79]:
```
1.707825127659933
```

In [80]:
```
np.sin(arr4)
```

Out[80]:
```
array([[-0.2794155 ,  0.6569866 ],
       [ 0.98935825,  0.41211849],
       [-0.54402111, -0.99999021]])
```

In [81]:
```
np.median(arr4)
```

Out[81]:
```
8.5
```

In [82]:
```
np.exp(arr4)
```

Out[82]:
```
array([[  403.42879349,  1096.63315843],
       [ 2980.95798704,  8103.08392758],
       [22026.46579481, 59874.1417152 ]])
```

reshape numpy arry

#revel #Reshape #Transpose #Stacking #Splitting

In [83]:
```
arr4
```

Out[83]:
```
array([[ 6,  7],
       [ 8,  9],
       [10, 11]])
```

In [84]:
```
arr4.ndim
```

Out[84]:
```
2
```

In [85]:
```
arr4.ravel()
```

Out[85]:
```
array([ 6,  7,  8,  9, 10, 11])
```

In [86]:
```
arr4.transpose()
```

Out[86]:
```
array([[ 6,  8, 10],
       [ 7,  9, 11]])
```

In [87]:
```
arr3
```

Out[87]:
```
array([[0, 1, 2],
       [3, 4, 5]])
```

In [88]:
```
arr5 = np.arange(12,18).reshape(2,3)
```

In [89]:
```
arr5
```

Out[89]:
```
array([[12, 13, 14],
       [15, 16, 17]])
```

In [91]:
```
np.hstack((arr3,arr5))
```

Out[91]:
```
array([[ 0,  1,  2, 12, 13, 14],
       [ 3,  4,  5, 15, 16, 17]])
```

In [93]:
```
np.vstack((arr3,arr5))
```

Out[93]:
```
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [12, 13, 14],
       [15, 16, 17]])
```

np.hsplit(arr3,3)

In [94]:
```
np.hsplit(arr3,3)
```

Out[94]:
```
[array([[0],
        [3]]),
 array([[1],
        [4]]),
 array([[2],
        [5]])]
```

In [95]:
```
np.vsplit(arr3,2)
```

Out[95]:
```
[array([[0, 1, 2]]), array([[3, 4, 5]])]
```

In [96]:
```
# fancy indexing
```

In [98]:
```
arr8 = np.arange(24).reshape(6,4)
```

In [99]:
```
arr8
```

Out[99]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15],
       [16, 17, 18, 19],
       [20, 21, 22, 23]])
```

In [100]:
```
arr8[[0,2,4]]
```

Out[100]:
```
array([[ 0,  1,  2,  3],
       [ 8,  9, 10, 11],
       [16, 17, 18, 19]])
```

In [104]:
```
# indexing with Boolean Arrays
```

In [101]:
```
arr = np.random.randint(low=1,high=100,size=20).reshape(4,5)
arr
```

Out[101]:
```
array([[21, 58, 79,  1, 76],
       [58, 20, 70, 93, 10],
       [18, 36, 56, 18, 78],
       [16, 52, 58, 85, 93]])
```

In [102]:
```
arr[0]
```

Out[102]:
```
array([21, 58, 79,  1, 76])
```

In [103]:
```
arr[0][2]
```

Out[103]:
```
79
```

In [105]:
```
arr>50
```

Out[105]:
```
array([[False,  True,  True, False,  True],
       [ True, False,  True,  True, False],
       [False, False,  True, False,  True],
       [False,  True,  True,  True,  True]])
```

In [107]:
```
(arr>50).shape
```

Out[107]:
```
(4, 5)
```

In [110]:
```
# indexing using Boolean array
arr[arr>50]
```

Out[110]:
```
array([58, 79, 76, 58, 70, 93, 56, 78, 52, 58, 85, 93])
```

In [111]:
```
arr[(arr>50)&(arr%2!=0)]
```

Out[111]:
```
array([79, 93, 85, 93])
```

In [112]:
```
arr[(arr>50)&(arr%2!=0)]=0
```

In [113]:
```
arr
```

Out[113]:
```
array([[21, 58,  0,  1, 76],
       [58, 20, 70,  0, 10],
       [18, 36, 56, 18, 78],
       [16, 52, 58,  0,  0]])
```

In [114]:
```
# plotting graphs using numpy
```

In [115]:
```
x = np.linspace(-40,40,100)
x
```

Out[115]:
```
array([-40.        , -39.19191919, -38.38383838, -37.57575758,
       -36.76767677, -35.95959596, -35.15151515, -34.34343434,
       -33.53535354, -32.72727273, -31.91919192, -31.11111111,
       -30.3030303 , -29.49494949, -28.68686869, -27.87878788,
       -27.07070707, -26.26262626, -25.45454545, -24.64646465,
       -23.83838384, -23.03030303, -22.22222222, -21.41414141,
       -20.60606061, -19.7979798 , -18.98989899, -18.18181818,
       -17.37373737, -16.56565657, -15.75757576, -14.94949495,
       -14.14141414, -13.33333333, -12.52525253, -11.71717172,
       -10.90909091, -10.1010101 ,  -9.29292929,  -8.48484848,
        -7.67676768,  -6.86868687,  -6.06060606,  -5.25252525,
        -4.44444444,  -3.63636364,  -2.82828283,  -2.02020202,
        -1.21212121,  -0.4040404 ,   0.4040404 ,   1.21212121,
         2.02020202,   2.82828283,   3.63636364,   4.44444444,
         5.25252525,   6.06060606,   6.86868687,   7.67676768,
         8.48484848,   9.29292929,  10.1010101 ,  10.90909091,
        11.71717172,  12.52525253,  13.33333333,  14.14141414,
        14.94949495,  15.75757576,  16.56565657,  17.37373737,
        18.18181818,  18.98989899,  19.7979798 ,  20.60606061,
        21.41414141,  22.22222222,  23.03030303,  23.83838384,
        24.64646465,  25.45454545,  26.26262626,  27.07070707,
        27.87878788,  28.68686869,  29.49494949,  30.3030303 ,
        31.11111111,  31.91919192,  32.72727273,  33.53535354,
        34.34343434,  35.15151515,  35.95959596,  36.76767677,
        37.57575758,  38.38383838,  39.19191919,  40.        ])
```

In [116]:
```
x.size
```

Out[116]:
```
100
```

In [117]:
```
y=np.sin(x)
```

In [118]:
```
y
```

Out[118]:
```
array([-0.74511316, -0.9969604 , -0.63246122,  0.12304167,  0.80247705,
        0.98580059,  0.55967698, -0.21245326, -0.85323945, -0.96653119,
       -0.48228862,  0.30011711,  0.89698277,  0.93931073,  0.40093277,
       -0.38531209, -0.93334716, -0.90436313, -0.31627868,  0.46733734,
        0.96203346,  0.86197589,  0.22902277, -0.54551809, -0.9828057 ,
       -0.81249769, -0.13988282,  0.61921119,  0.995493  ,  0.75633557,
        0.04959214, -0.68781042, -0.99999098, -0.69395153,  0.0411065 ,
        0.75075145,  0.99626264,  0.62585878, -0.13146699, -0.8075165 ,
       -0.98433866, -0.55261747,  0.22074597,  0.85763861,  0.96431712,
        0.47483011, -0.30820902, -0.90070545, -0.93636273, -0.39313661,
        0.39313661,  0.93636273,  0.90070545,  0.30820902, -0.47483011,
       -0.96431712, -0.85763861, -0.22074597,  0.55261747,  0.98433866,
        0.8075165 ,  0.13146699, -0.62585878, -0.99626264, -0.75075145,
       -0.0411065 ,  0.69395153,  0.99999098,  0.68781042, -0.04959214,
       -0.75633557, -0.995493  , -0.61921119,  0.13988282,  0.81249769,
        0.9828057 ,  0.54551809, -0.22902277, -0.86197589, -0.96203346,
       -0.46733734,  0.31627868,  0.90436313,  0.93334716,  0.38531209,
       -0.40093277, -0.93931073, -0.89698277, -0.30011711,  0.48228862,
        0.96653119,  0.85323945,  0.21245326, -0.55967698, -0.98580059,
       -0.80247705, -0.12304167,  0.63246122,  0.9969604 ,  0.74511316])
```

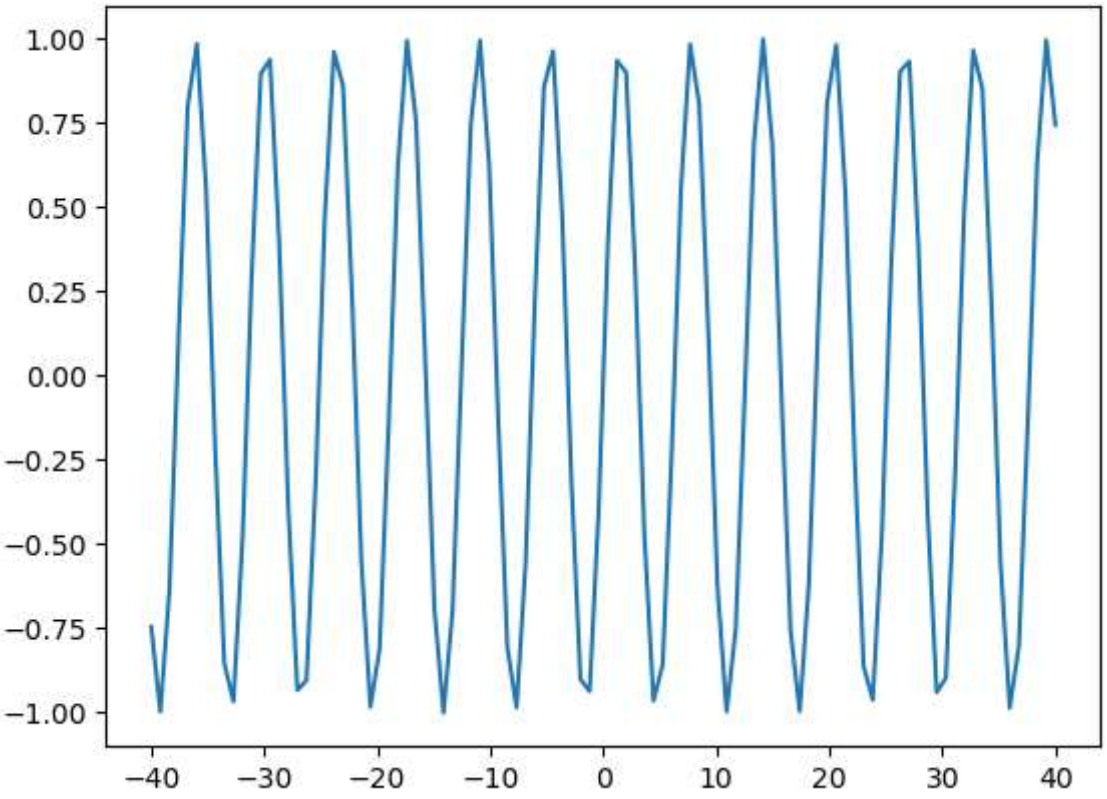
In [119]:
```
y.size
```

Out[119]:
```
100
```

In [121]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [122]:
```
plt.plot(x,y)
```

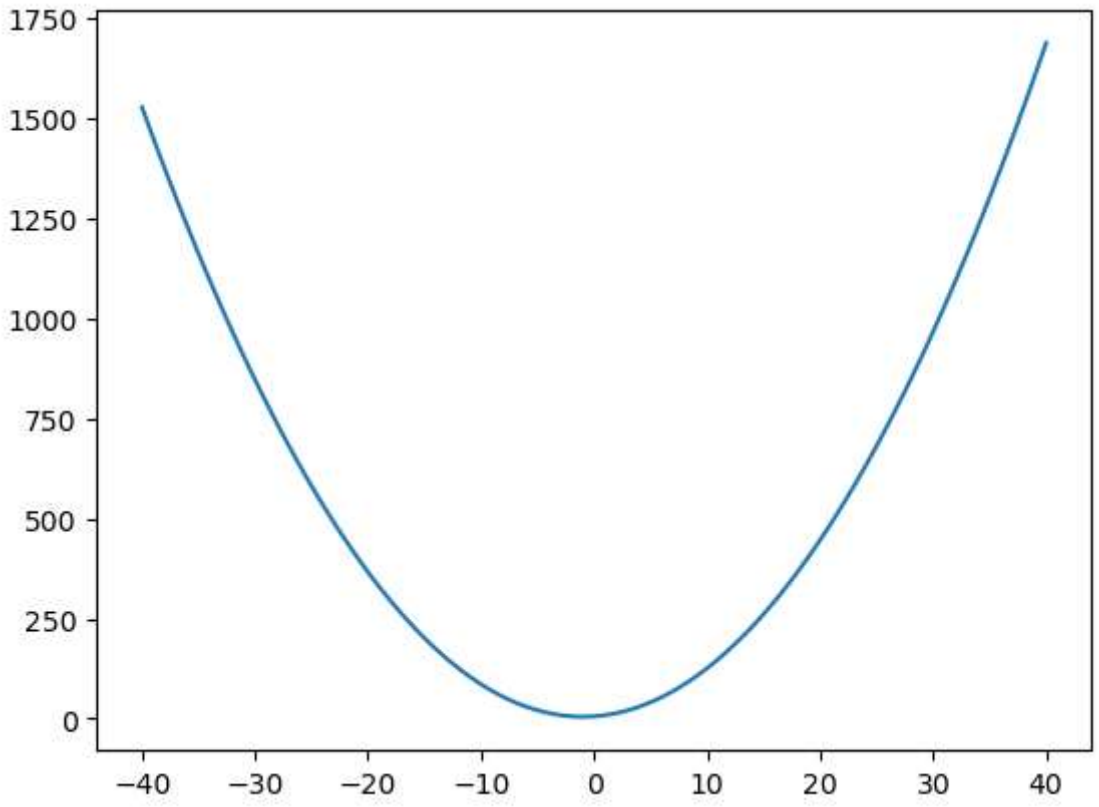
Out[122]:
```
[<matplotlib.lines.Line2D at 0x22131ceb8e0>]
```



In [123]:
```
y = x*x+2*x+6
```

In [124]:
```
plt.plot(x,y)
```

Out[124]:
```
[<matplotlib.lines.Line2D at 0x22131f78c40>]
```



In [125]:
```
# scenario 1
a1 = np.arange(8).reshape(2,4)
a2 = np.arange(8,16).reshape(2,4)

print(a1)
print(a2)
```

```
[[0 1 2 3]
 [4 5 6 7]]
[[ 8  9 10 11]
 [12 13 14 15]]
```

In [126]:
```
a1+a2
```

Out[126]:
```
array([[ 8, 10, 12, 14],
       [16, 18, 20, 22]])
```

In [127]:
```
# scenario 2
a3 = np.arange(9).reshape(3,3)
a4 = np.arange(3).reshape(1,3)


print(a3,a4)
```

```
[[0 1 2]
 [3 4 5]
 [6 7 8]] [[0 1 2]]
```

In [128]:
```
a3 +a4
```

Out[128]:
```
array([[ 0,  2,  4],
       [ 3,  5,  7],
       [ 6,  8, 10]])
```

```
# Broadcasting
#the term broadcasting refers to the ability of numpy to treat arrays of different shapes durings arthimatic operation arthimatic operation on array are usually done on corresponding elements if two array are of exactly the same shape then these operation are smoothly performed
# if the dimension of two arrays are dissimilar , elements operations are not possible however operation on dissimilier array is still possible in numpy because of the broadcasting capability . the smaller array array is broadcast is broadcast to the size of the larger array so that they have compatibility shapes
```

In [1]:
```
# Rules of Broadcasting
```

In [4]:
```
# if x = m and y = n , operation will take place.
a1 = np.arange(8).reshape(2,4)
a2 = np.arange(8,16).reshape(2,4)

a1+a2
```

Out[4]:
```
array([[ 8, 10, 12, 14],
       [16, 18, 20, 22]])
```

In [7]:
```
# If x=1 and y=n then also operation will take place(same dimension)
a5 = np.arange(3).reshape(1,3)
a6 = np.arange(12).reshape(4,3)

print(a5)
print(a6)
```

```
[[0 1 2]]
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

In [6]:
```
a5+a6
```

Out[6]:
```
array([[ 0,  2,  4],
       [ 3,  5,  7],
       [ 6,  8, 10],
       [ 9, 11, 13]])
```

In [8]:
```
# if x=1 and y!=n then also operation will not take place

a9 =np.arange(3).reshape(1,3)
a10 = np.arange(16).reshape(4,4)

print(a9)
print(a10)
```

```
[[0 1 2]]
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [12 13 14 15]]
```

In [10]:
```
a9 + a10
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[10], line 1
----> 1 a9 + a10

ValueError: operands could not be broadcast together with shapes (1,3) (4,4)
```

In [17]:
```
# if x = 1 and y = 1 then the operation will take place no matter what

    a13 =np.arange(1).reshape(1,1)
    a14 = np.arange(20).reshape(4,5)

    print(13)
    print(a14)
```

```
  Cell In[17], line 3
    a13 =np.arange(1).reshape(1,1)
    ^
IndentationError: unexpected indent
```

In [18]:
```
a13 + a14
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[18], line 1
----> 1 a13 + a14

NameError: name 'a13' is not defined
```

In [19]:
```
# if they are of different dimensions

a15 = np.arange(4)
a16 = np.arange(20).reshape(5,4)

print(a15)
print(a16)
```

```
[0 1 2 3]
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [12 13 14 15]
 [16 17 18 19]]
```

In [20]:
```
a15+a16
```

Out[20]:
```
array([[ 0,  2,  4,  6],
       [ 4,  6,  8, 10],
       [ 8, 10, 12, 14],
       [12, 14, 16, 18],
       [16, 18, 20, 22]])
```

In [21]:
```
# important function in numpy.
```

In [24]:
```
np.random.ramdom()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[24], line 1
----> 1 np.random.ramdom()

AttributeError: module 'numpy.random' has no attribute 'ramdom'
```

In [27]:
```
np.random.seed(1)
np.random.random()
```

Out[27]:
```
0.417022004702574
```

In [30]:
```
np.random.uniform(3,10)
```

Out[30]:
```
5.116328008422879
```

In [31]:
```
np.random.uniform(1,100,10)
```

Out[31]:
```
array([15.52883319, 10.14152088, 19.43976093, 35.21051198, 40.27997995,
       54.34285667, 42.50025693, 68.83673054, 21.24077272, 87.9336262 ])
```

In [32]:
```
np.random.uniform(1,100,10).reshape(2,5)
```

Out[32]:
```
array([[ 3.71137173, 67.37628351, 42.31317543, 56.31029302, 14.89830692],
       [20.61204742, 80.2737123 , 96.857896  , 32.02899364, 69.53993895]])
```

In [35]:
```
np.random.randint(1,10)
```

Out[35]:
```
2
```

In [36]:
```
np.random.randint(1,10,5)
```

Out[36]:
```
array([9, 9, 4, 9, 8])
```

In [37]:
```
np.random.randint(1,10,15).reshape(3,5)
```

Out[37]:
```
array([[4, 7, 6, 2, 4],
       [5, 9, 2, 5, 1],
       [4, 3, 1, 5, 3]])
```

In [41]:
```
a = np.random.randint(1,10,6)
a
```

Out[41]:
```
array([8, 5, 6, 4, 7, 9])
```

In [42]:
```
np.max(a)
```

Out[42]:
```
9
```

In [43]:
```
np.min(a)
```

Out[43]:
```
4
```

In [46]:
```
a[np.argmax(a)]
```

Out[46]:
```
9
```

In [47]:
```
a[np.argmin(a)]
```

Out[47]:
```
4
```

In [48]:
```
np.argmin(a)
```

Out[48]:
```
3
```

In [49]:
```
a = np.random.randint(1,10,6)
```

In [50]:
```
a
```

Out[50]:
```
array([4, 1, 9, 8, 8, 2])
```

In [51]:
```
a[a%2==1]=-1
a
```

Out[51]:
```
array([ 4, -1, -1,  8,  8,  2])
```

In [54]:
```
#where (condition ,ture,false)
```

In [55]:
```
a = np.random.randint(1,50,6)
```

In [56]:
```
a
```

Out[56]:
```
array([14,  4,  1, 14,  7, 22])
```

In [57]:
```
np.where (a%2==1,-1,a)
```

Out[57]:
```
array([14,  4, -1, 14, -1, 22])
```

In [58]:
```
a
```

Out[58]:
```
array([14,  4,  1, 14,  7, 22])
```

In [59]:
```
out = np.where(a%2==1,-1,a)
out
```

Out[59]:
```
array([14,  4, -1, 14, -1, 22])
```

In [60]:
```
a= np.random.randint(1,50,10)
```

In [61]:
```
a
```

Out[61]:
```
array([ 7,  3, 13, 28, 22, 12,  8, 14,  9, 12])
```

In [63]:
```
a = np.sort(a)
```

In [64]:
```
a
```

Out[64]:
```
array([ 3,  7,  8,  9, 12, 12, 13, 14, 22, 28])
```

In [65]:
```
np.percentile(a,25)
```

Out[65]:
```
8.25
```

In [66]:
```
np.percentile(a,50)
```

Out[66]:
```
12.0
```

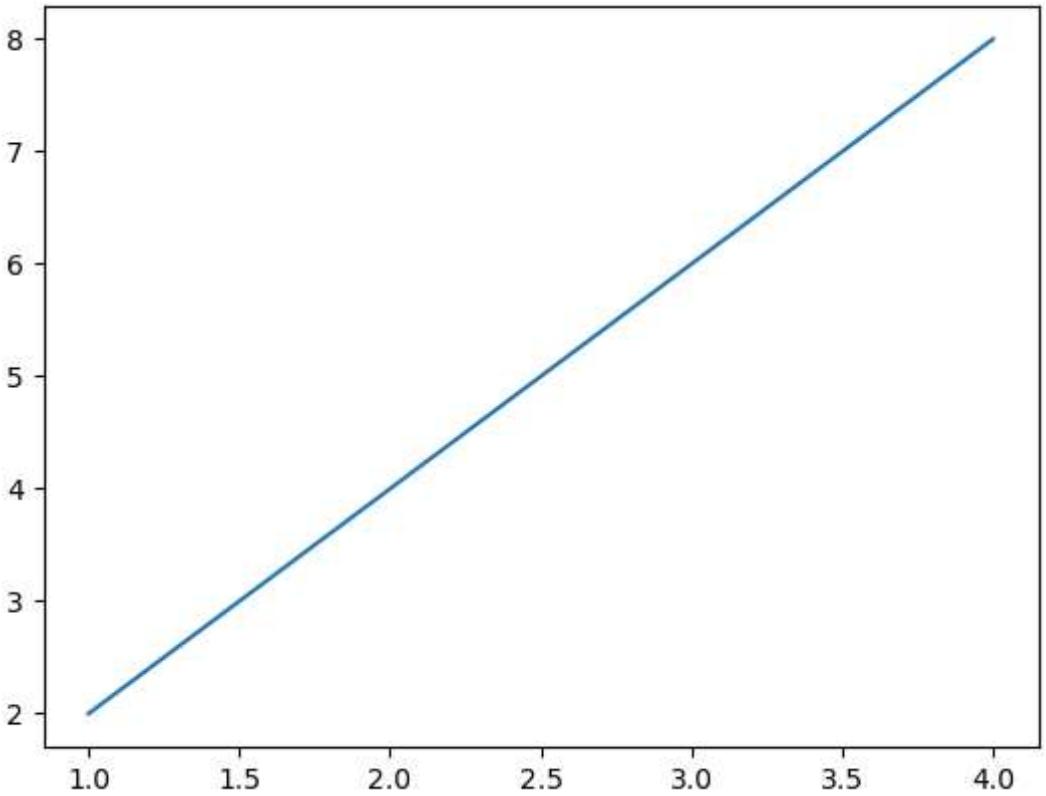
In [67]:
```
np.percentile(a,99.8)
```

Out[67]:
```
27.891999999999996
```

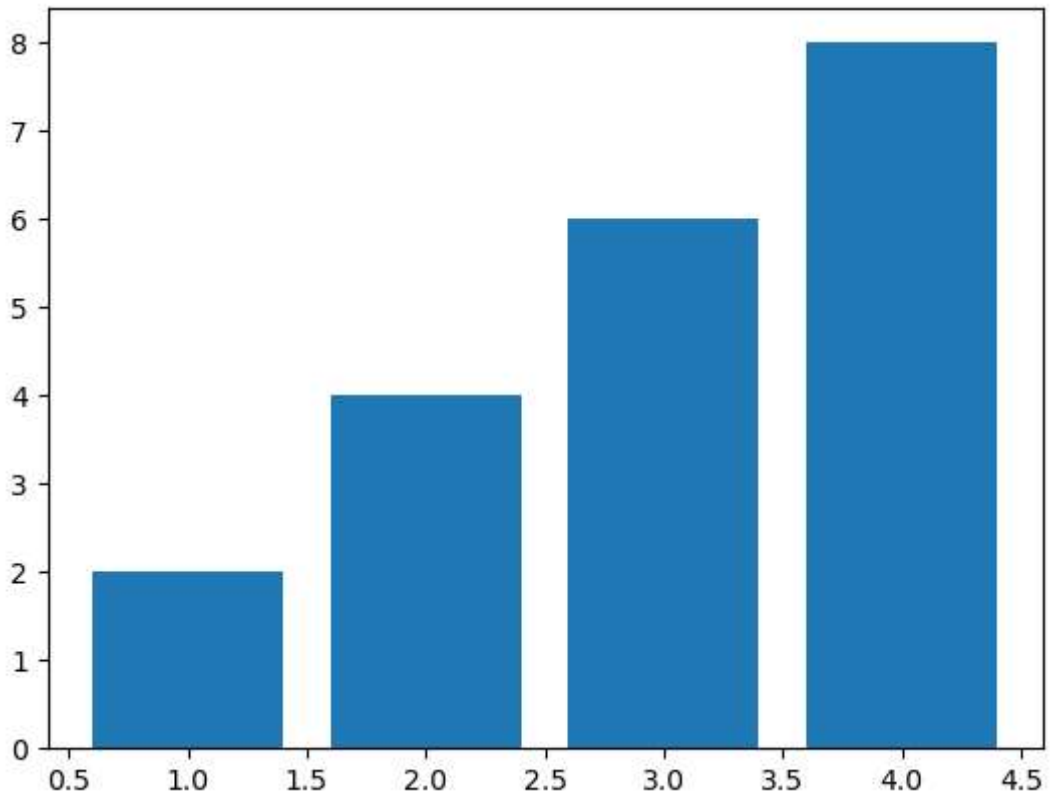
matplotlib

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

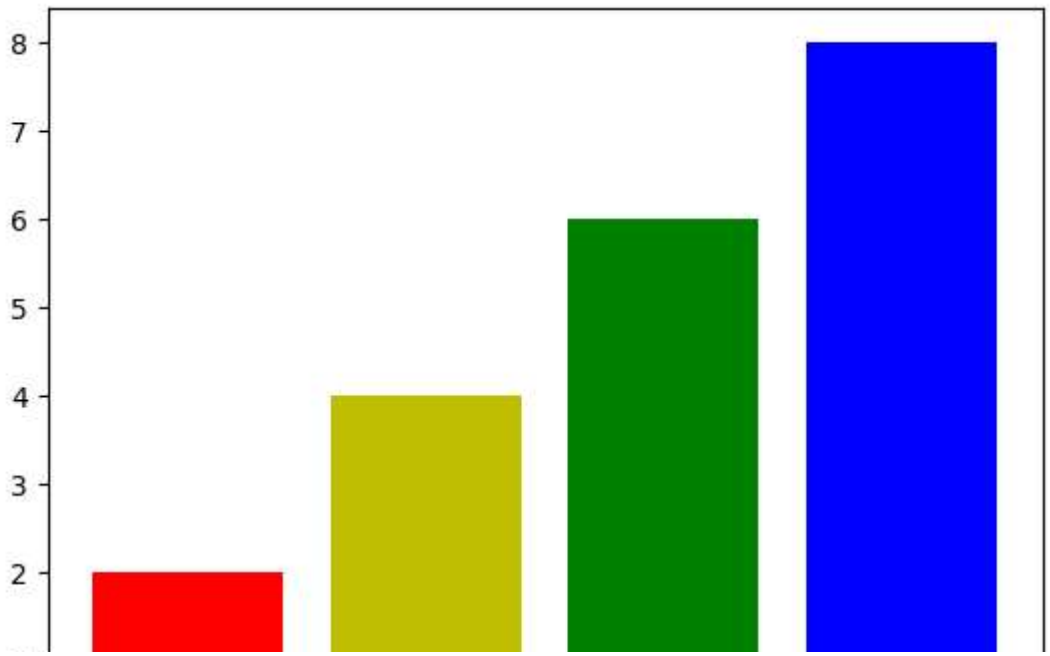
In [5]:
```
x = [1,2,3,4]
y = [2,4,6,8]
plt.plot(x,y)
plt.show()
```



In [6]:
```
x = [1,2,3,4]
y = [2,4,6,8]
plt.bar(x,y)
plt.show()
```

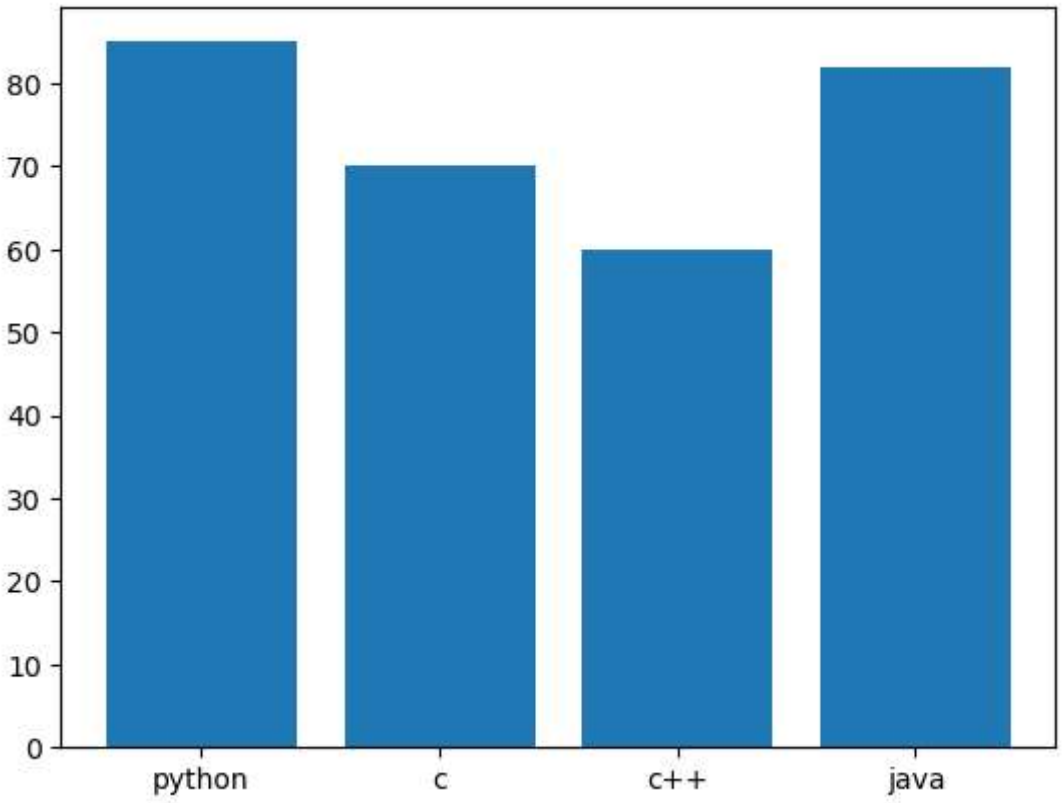


In [10]:
```
x = [1,2,3,4]
y = [2,4,6,8]
c=["r",'y',"g",'b']
plt.bar(x,y,color =c)
plt.show()
```

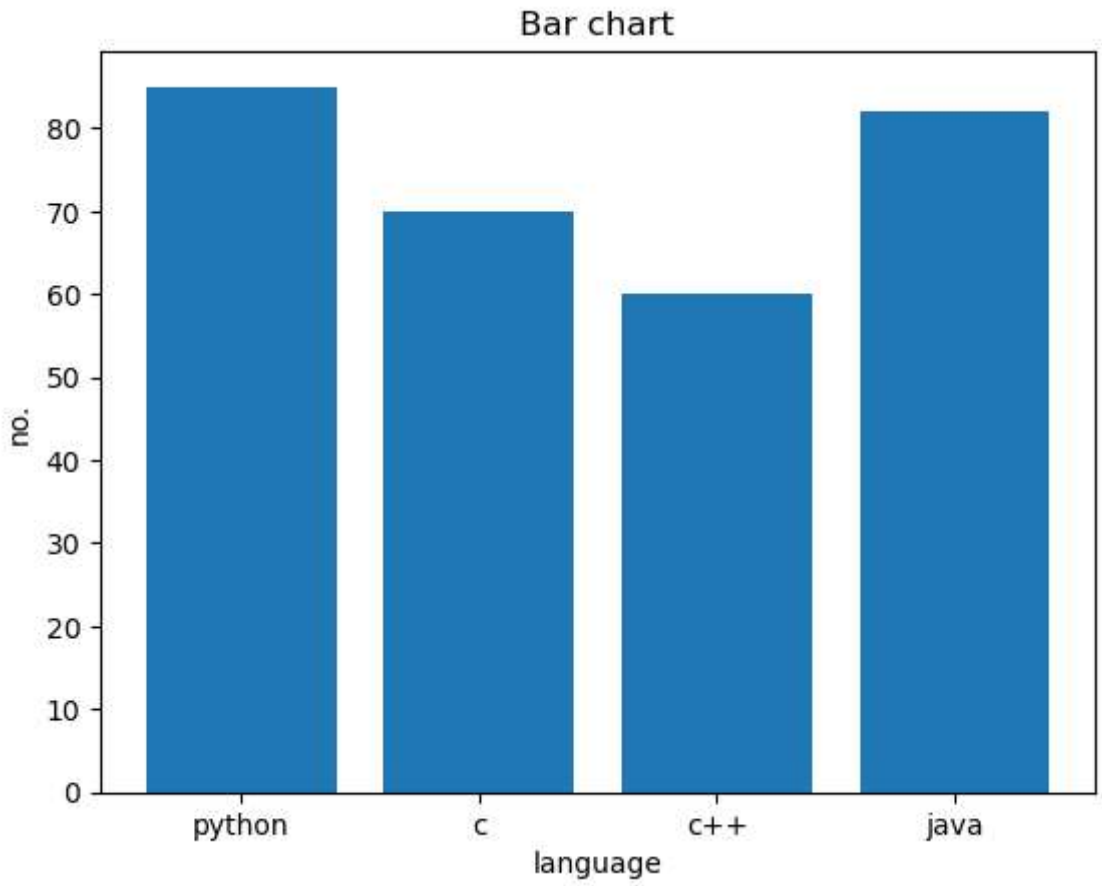


Bar plot

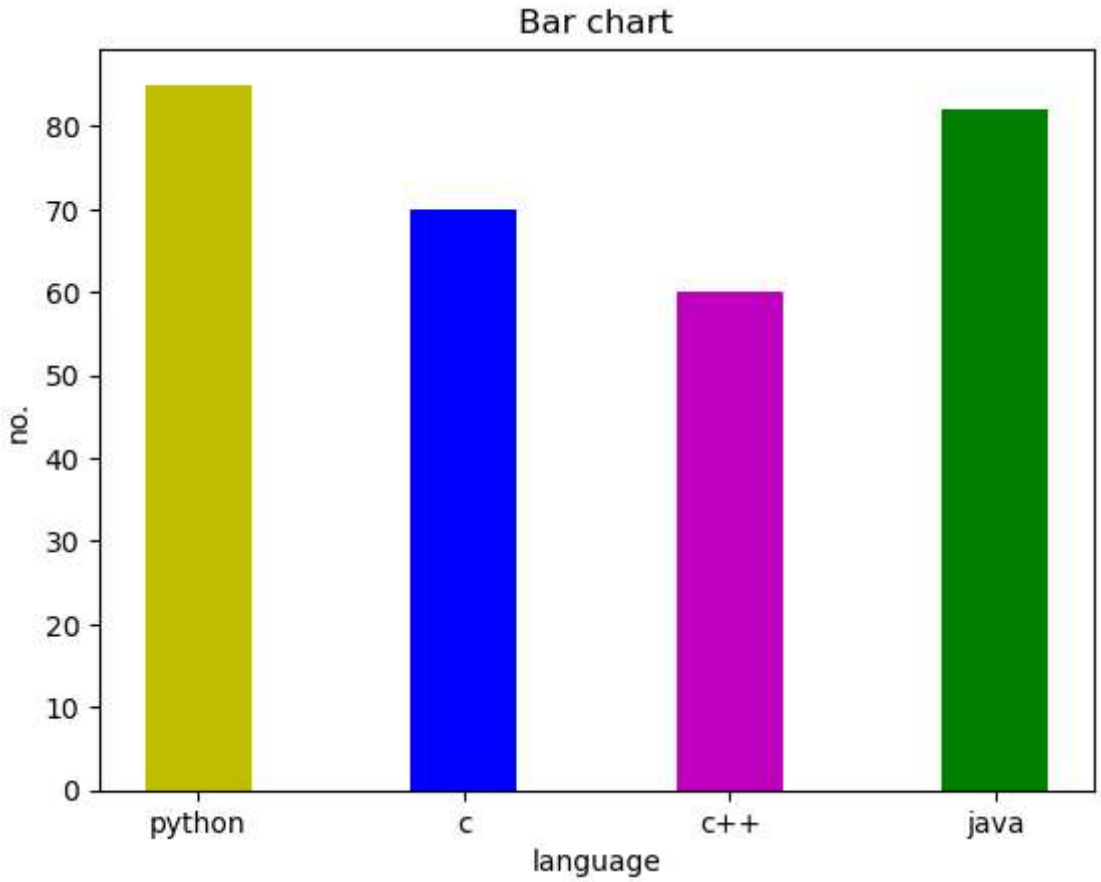
In [11]:
```
x = ['python','c','c++','java']
y = [85,70,60,82]
plt.bar(x,y)
plt.show()
```



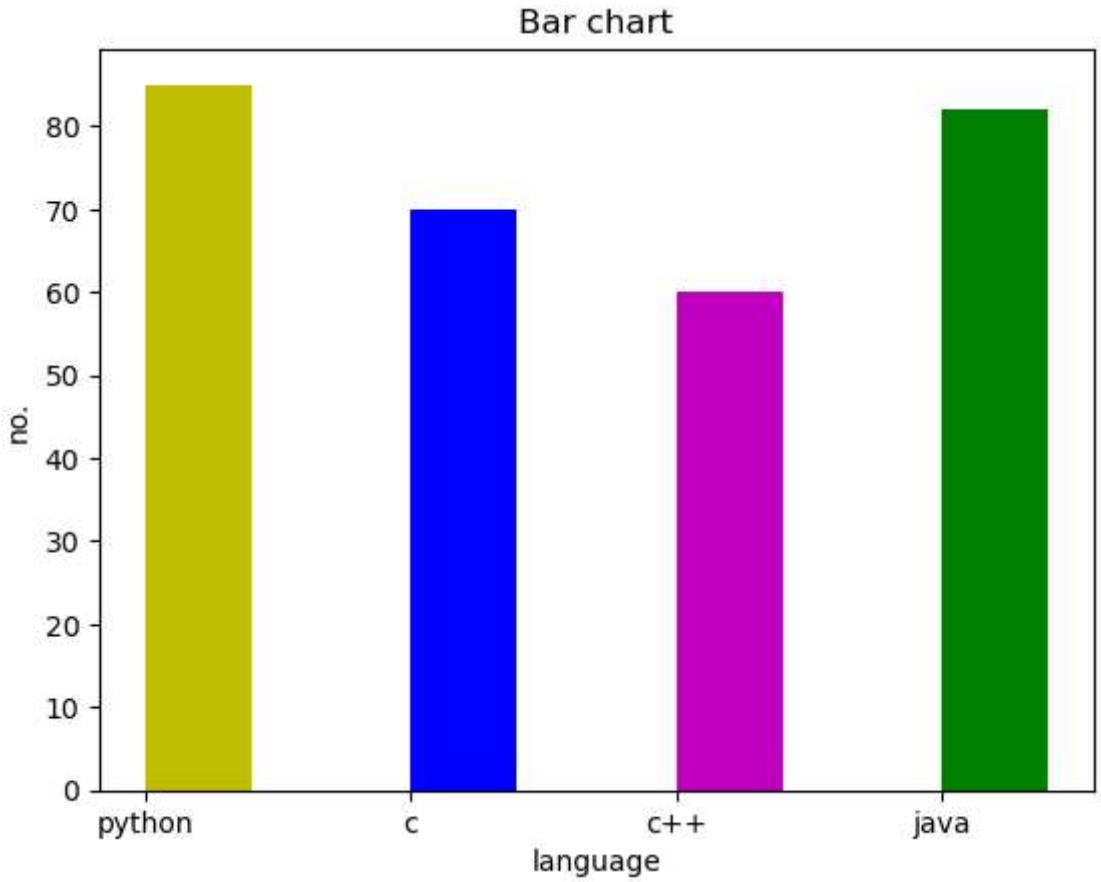
In [18]:
```
x = ['python','c','c++','java']
y = [85,70,60,82]
plt.xlabel("language")
plt.ylabel("no.")
plt.title('Bar chart')
plt.bar(x,y)
plt.show()
```



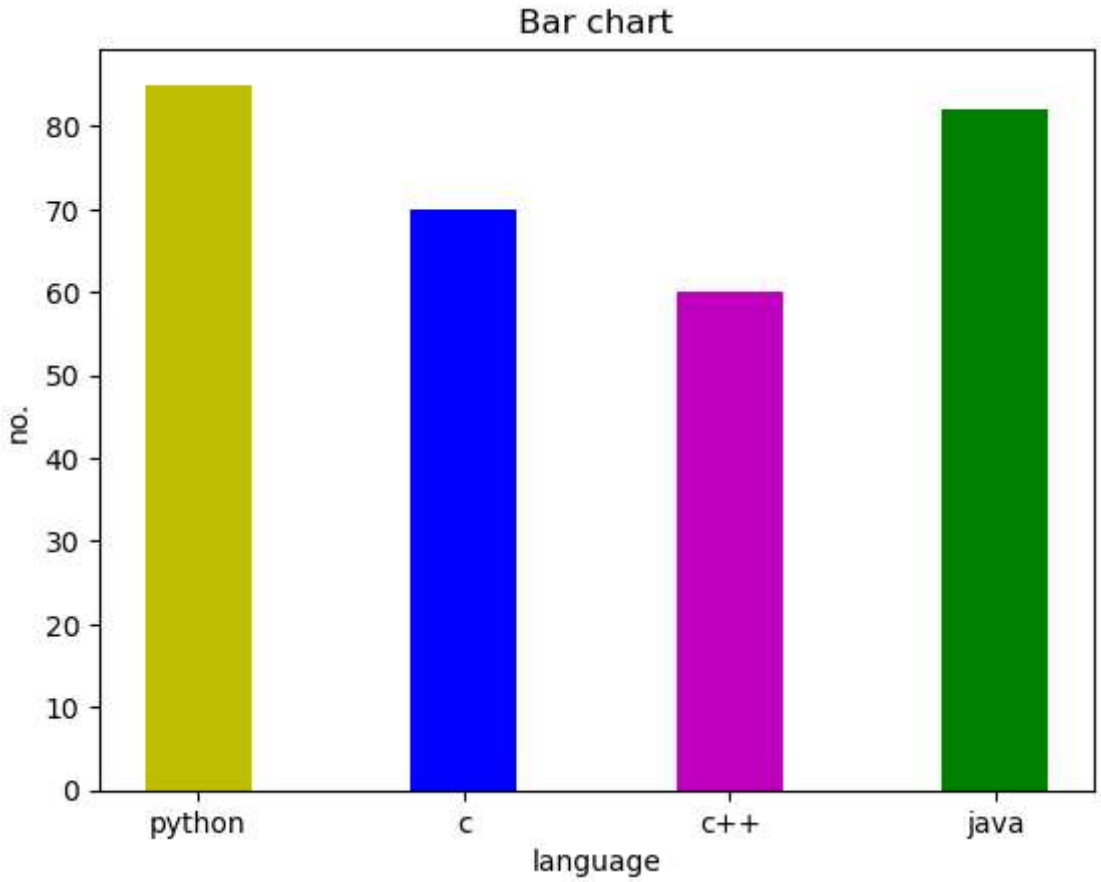
In [21]:
```
x = ['python','c','c++','java']
y = [85,70,60,82]
plt.xlabel("language")
plt.ylabel("no.")
plt.title('Bar chart')
c=['y','b','m','g']
plt.bar(x,y,width=0.4,color=c)
plt.show()
```



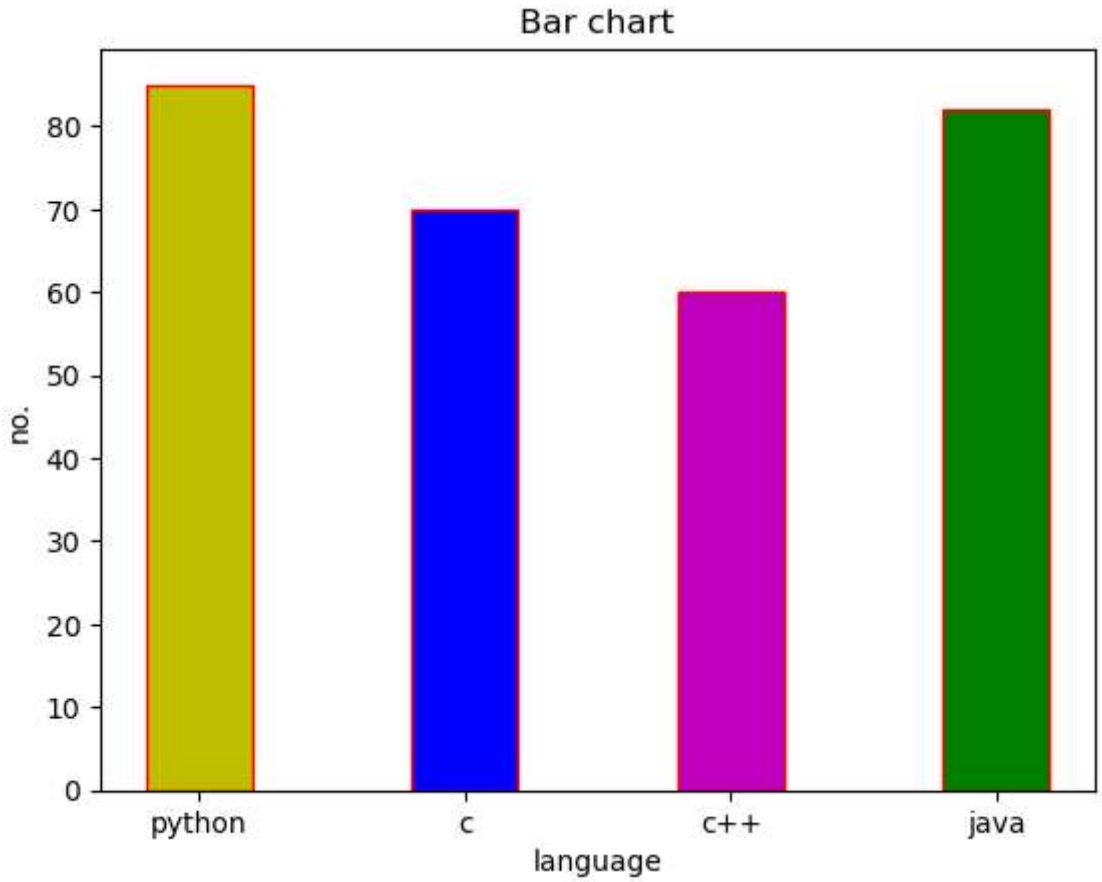
In [22]:
```
x = ['python','c','c++','java']
y = [85,70,60,82]
plt.xlabel("language")
plt.ylabel("no.")
plt.title('Bar chart')
c=['y','b','m','g']
plt.bar(x,y,width=0.4,color=c,align = 'edge')
plt.show()
```



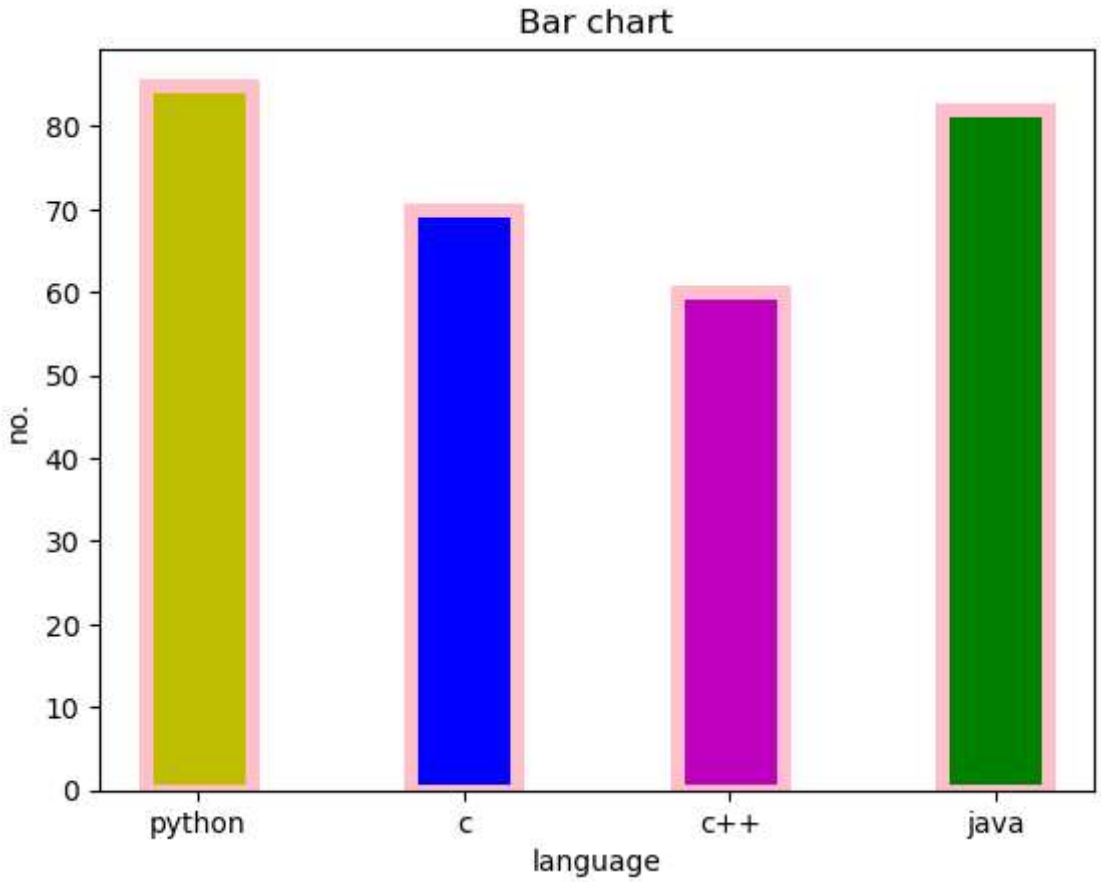
In [23]:
```
x = ['python','c','c++','java']
y = [85,70,60,82]
plt.xlabel("language")
plt.ylabel("no.")
plt.title('Bar chart')
c=['y','b','m','g']
plt.bar(x,y,width=0.4,color=c,align = 'center')
plt.show()
```



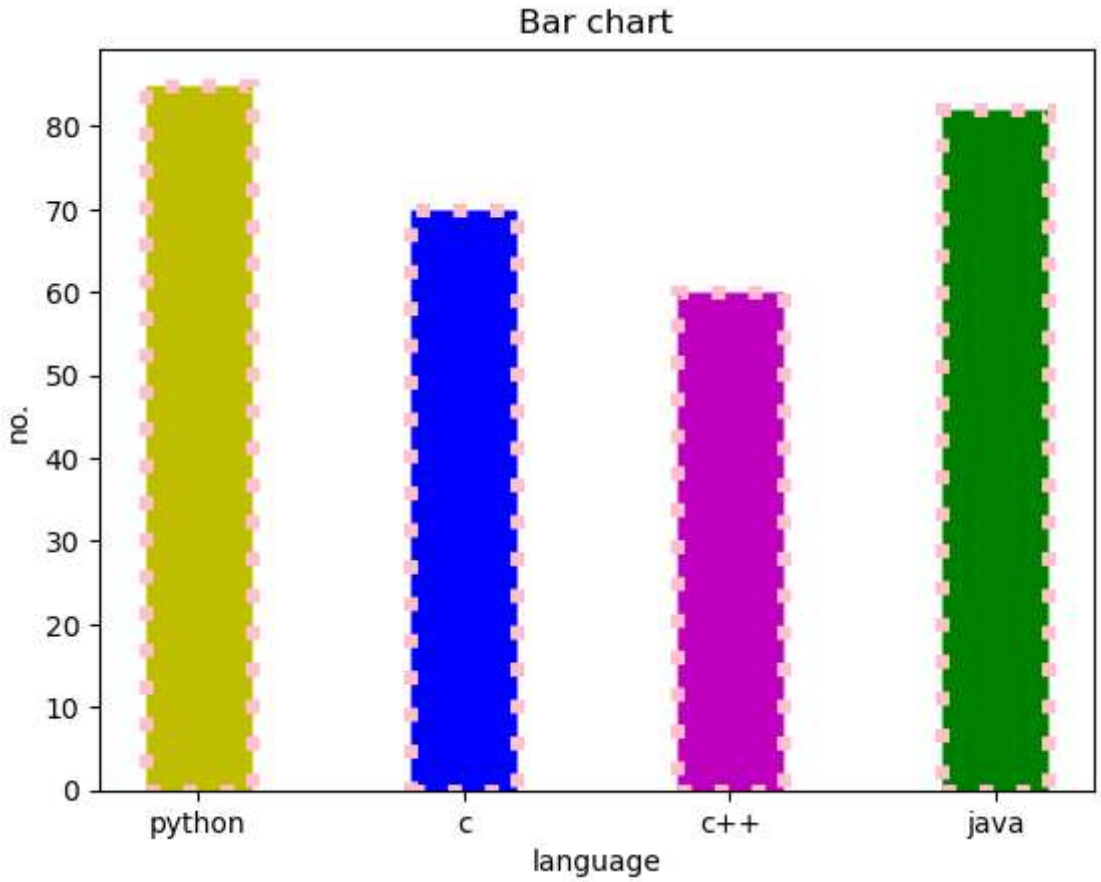
In [28]:
```
x = ['python','c','c++','java']
y = [85,70,60,82]
plt.xlabel("language")
plt.ylabel("no.")
plt.title('Bar chart')
c=['y','b','m','g']
plt.bar(x,y,width=0.4,color=c,edgecolor='r')
plt.show()
```



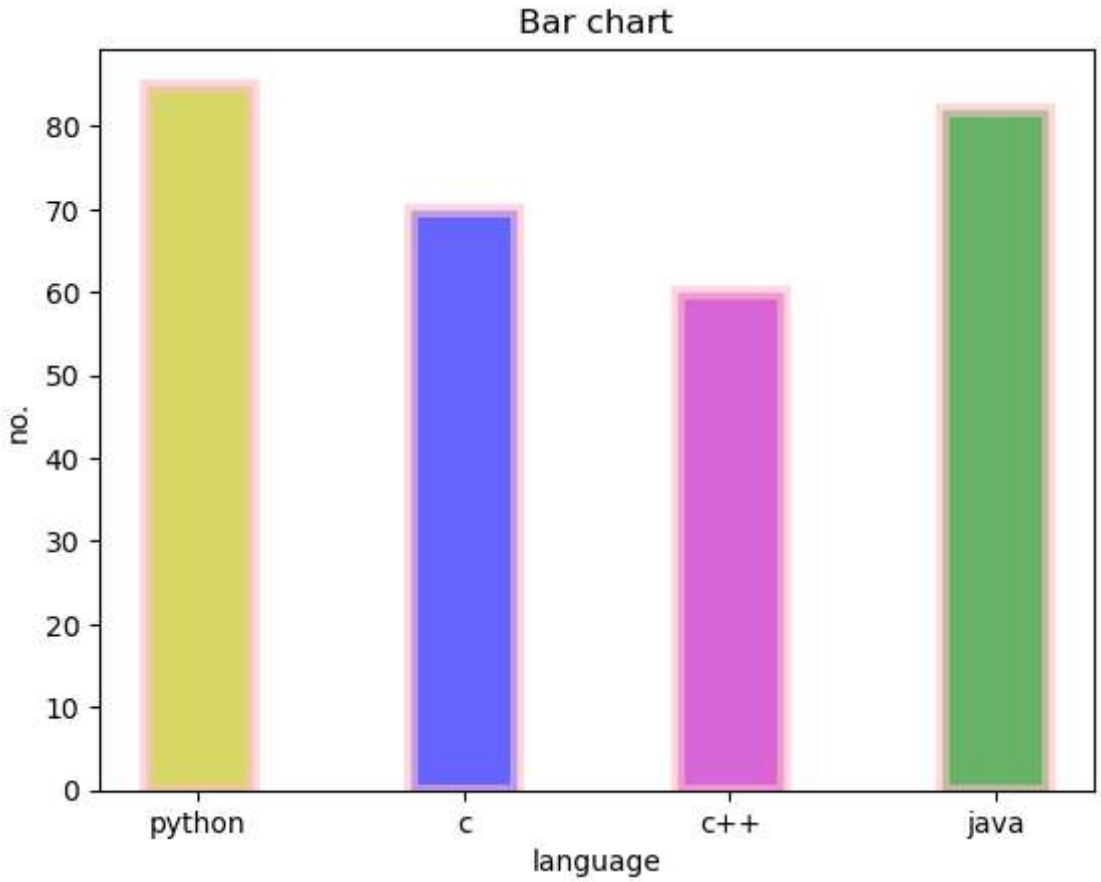
In [30]:
```
x = ['python','c','c++','java']
y = [85,70,60,82]
plt.xlabel("language")
plt.ylabel("no.")
plt.title('Bar chart')
c=['y','b','m','g']
plt.bar(x,y,width=0.4,color=c,edgecolor='pink',linewidth=5)
plt.show()
```



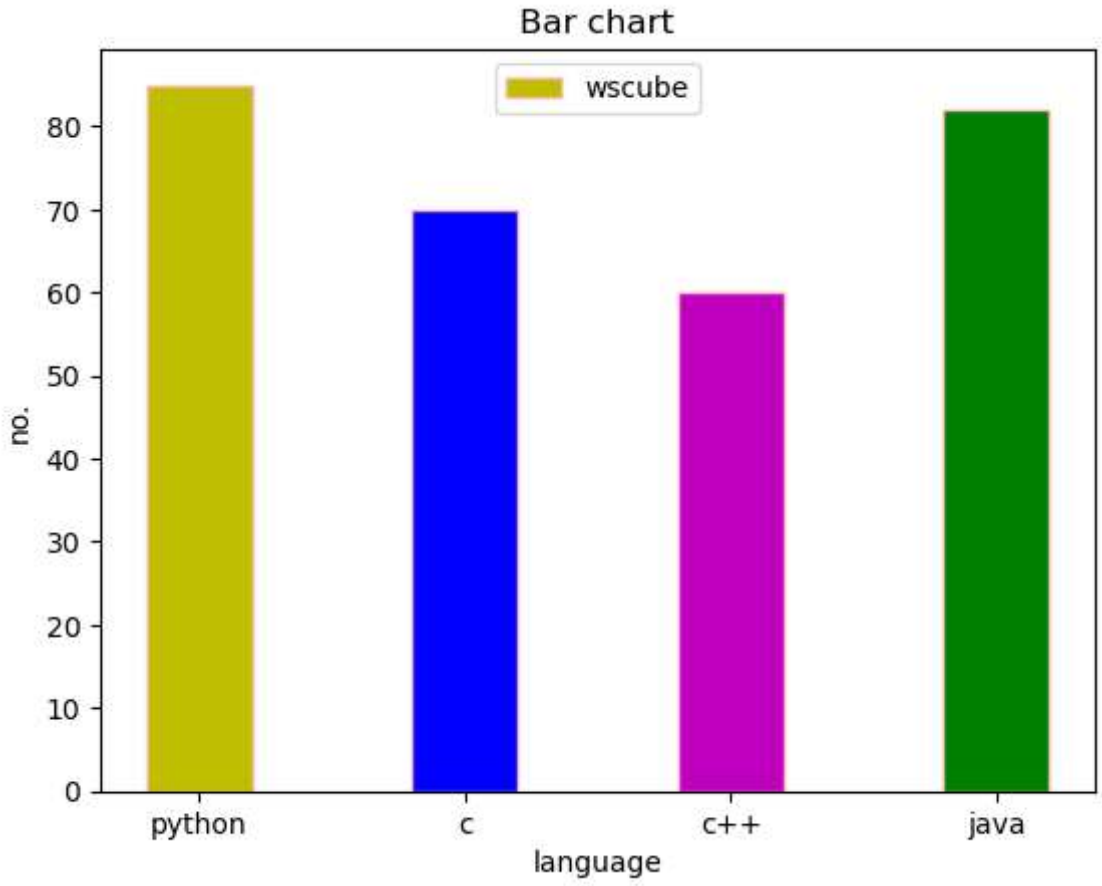
In [32]:
```
x = ['python','c','c++','java']
y = [85,70,60,82]
plt.xlabel("language")
plt.ylabel("no.")
plt.title('Bar chart')
c=['y','b','m','g']
plt.bar(x,y,width=0.4,color=c,edgecolor='pink',linewidth=5,linestyle=":")
plt.show()
```



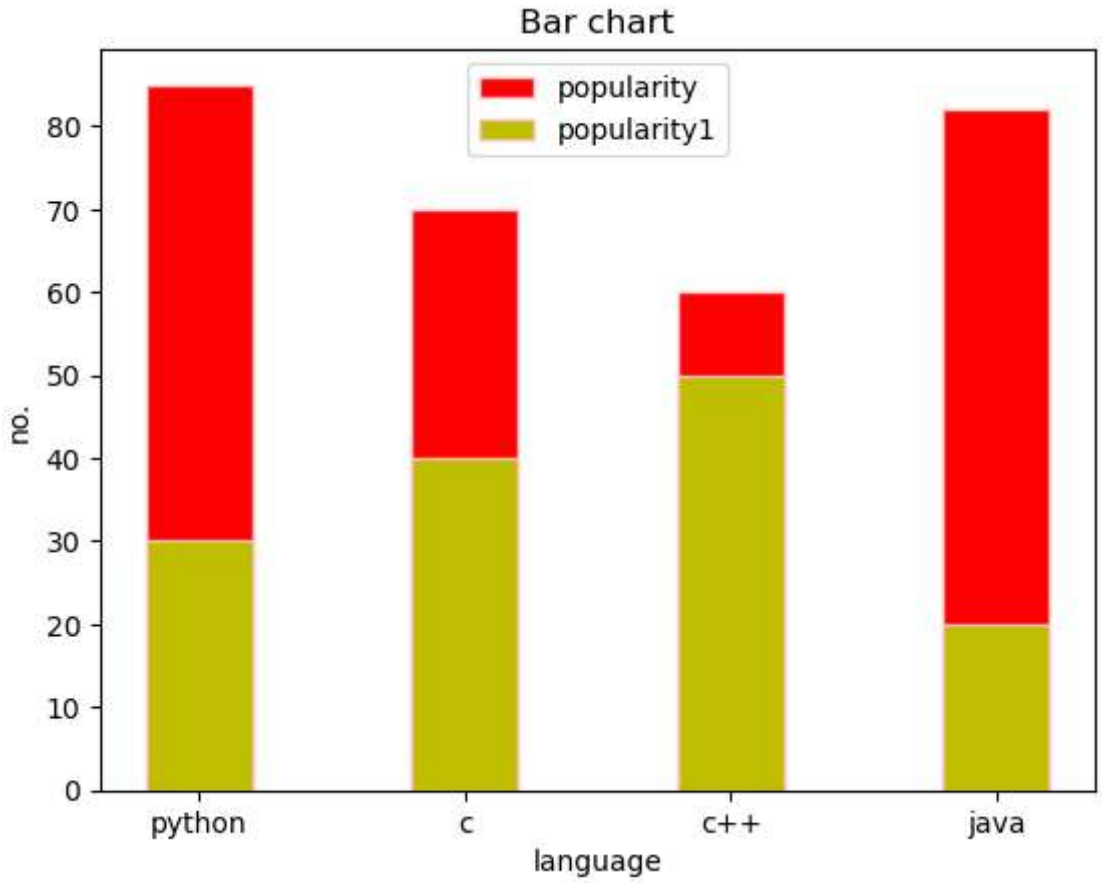
In [34]:
```
x = ['python','c','c++','java']
y = [85,70,60,82]
plt.xlabel("language")
plt.ylabel("no.")
plt.title('Bar chart')
c=['y','b','m','g']
plt.bar(x,y,width=0.4,color=c,edgecolor='pink',linewidth=5,alpha = 0.6)
plt.show()
```



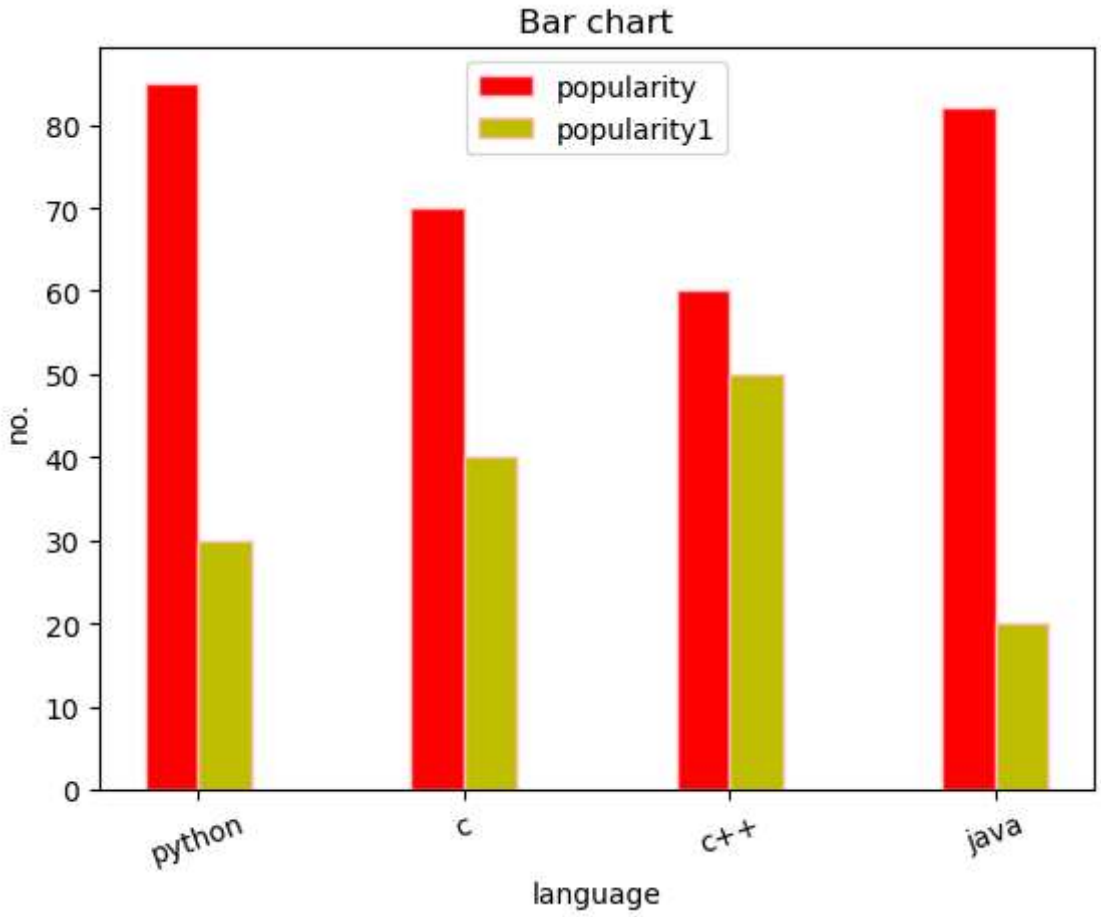
In [36]:
```
x = ['python','c','c++','java']
y = [85,70,60,82]
plt.xlabel("language")
plt.ylabel("no.")
plt.title('Bar chart')
c=['y','b','m','g']
plt.bar(x,y,width=0.4,color=c,edgecolor='pink',label='wscube')
plt.legend()
plt.show()
```



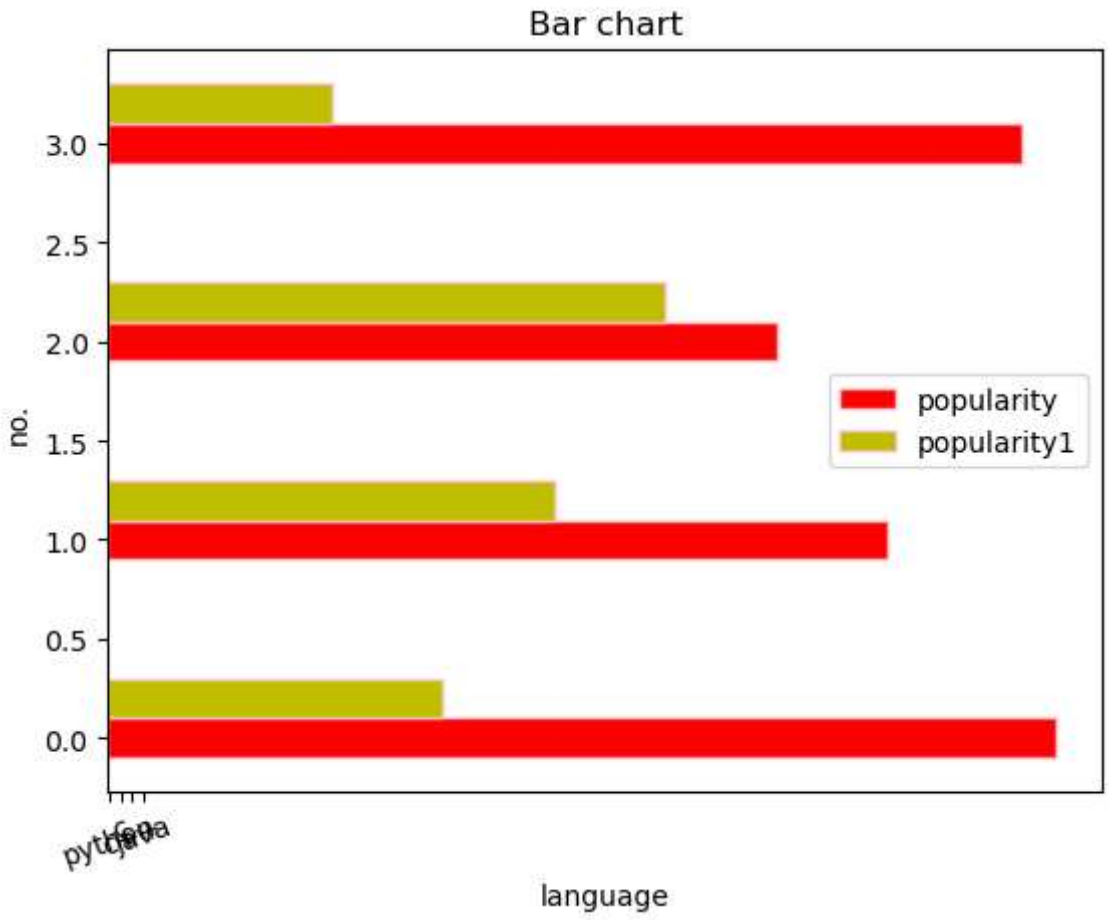
In [3]:
```
x = ['python','c','c++','java']
y = [85,70,60,82]
z = [30,40,50,20]
plt.xlabel("language")
plt.ylabel("no.")
plt.title('Bar chart')
plt.bar(x,y,width=0.4,color="red",edgecolor='pink',label='popularity')
plt.bar(x,z,width=0.4,color="y",edgecolor='pink',label='popularity1')
plt.legend()
plt.show()
```



In [11]:
```
import numpy as np
x = ['python','c','c++','java']
y = [85,70,60,82]
z = [30,40,50,20]
width = 0.2
p = np.arange(len(x))
p1 = [j+width for j in p]

plt.xlabel("language")
plt.ylabel("no.")
plt.title('Bar chart')

plt.bar(p,y,width,color="red",edgecolor='pink',label='popularity')
plt.bar(p1,z,width,color="y",edgecolor='pink',label='popularity1')

plt.xticks(p+width/2,x,rotation=20)
plt.legend()
plt.show()
```

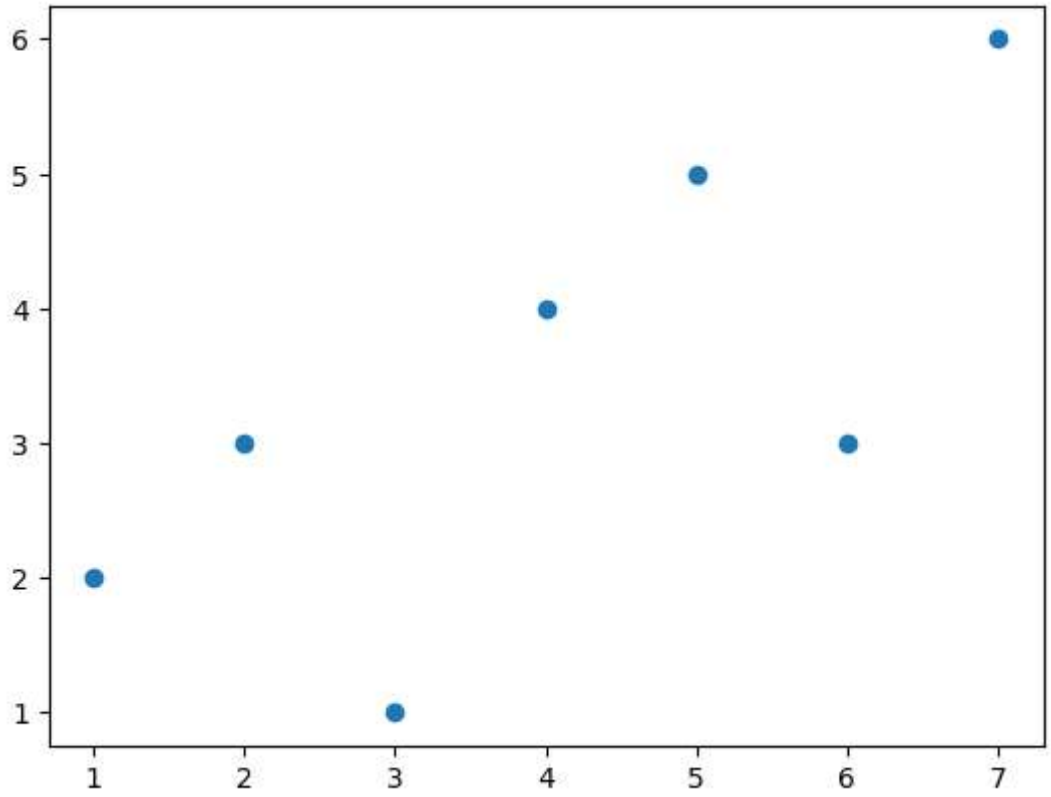


In [17]:
```
x = ['python','c','c++','java']
y = [85,70,60,82]
z = [30,40,50,20]
width = 0.2
p = np.arange(len(x))
p1 = [j+width for j in p]

plt.xlabel("language")
plt.ylabel("no.")
plt.title('Bar chart')

plt.barh(p,y,width,color="red",edgecolor='pink',label='popularity')
plt.barh(p1,z,width,color="y",edgecolor='pink',label='popularity1')

plt.xticks(p+width/2,x,rotation=20)
plt.legend()
plt.show()
```

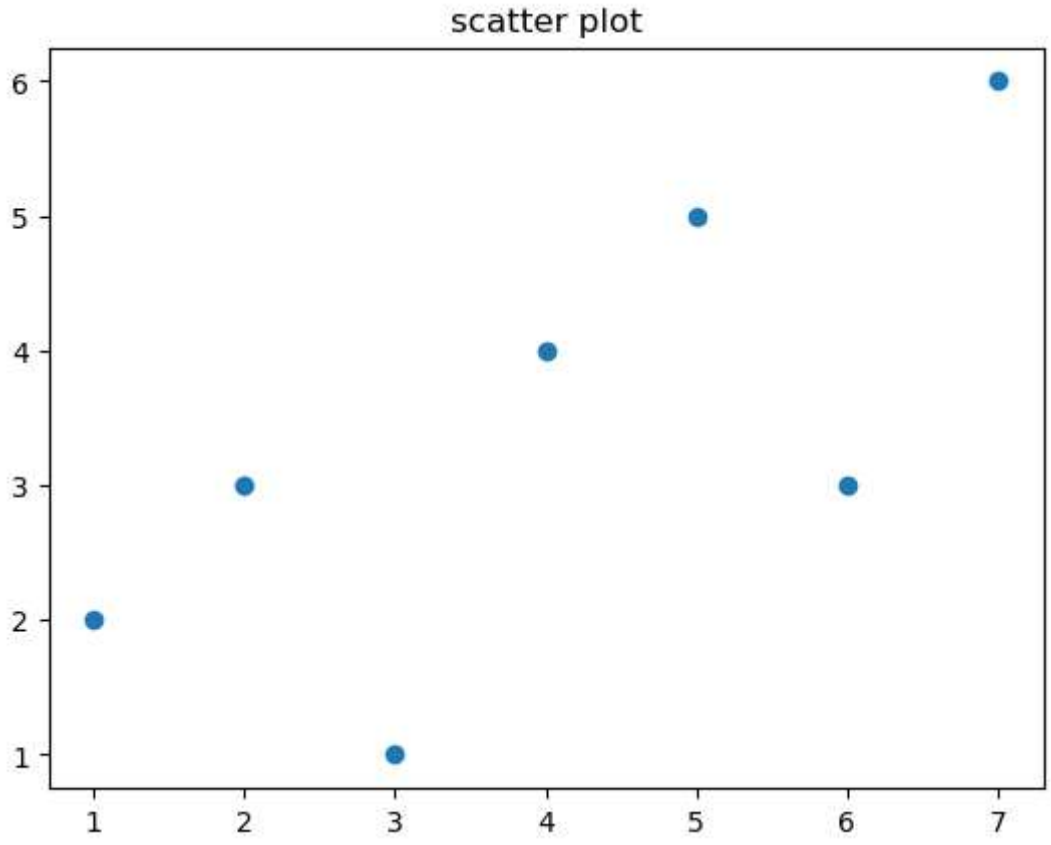


scatter plot

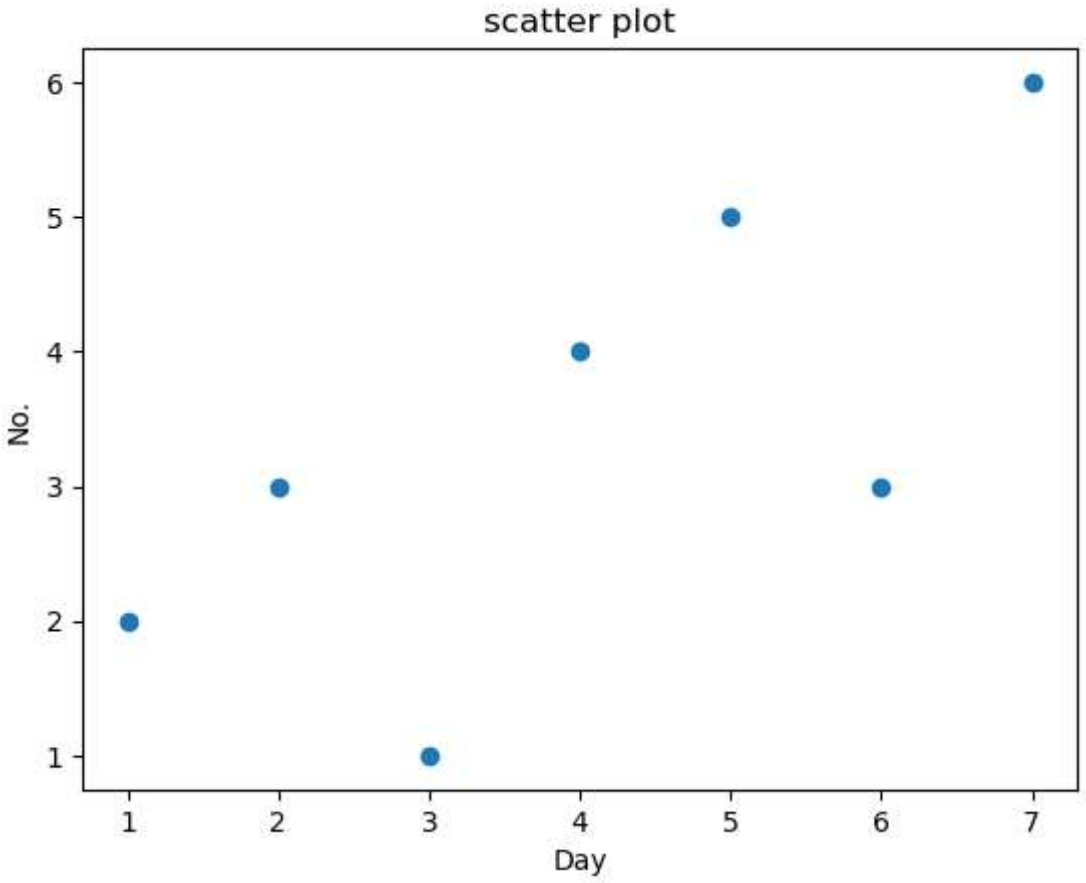
In [18]:
```
x = [1,2,3,4,5,6,7]
y=[2,3,1,4,5,3,6]
plt.scatter(x,y)
plt.show()
```



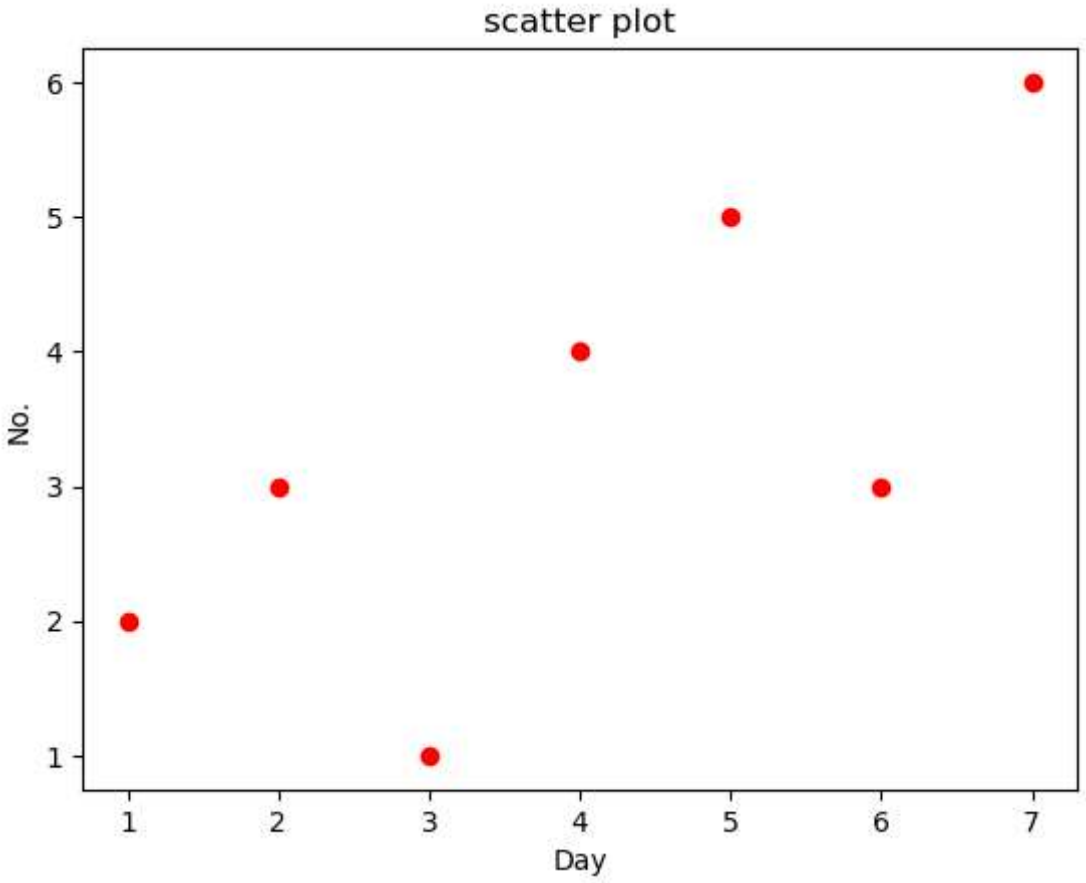
In [20]:
```
x = [1,2,3,4,5,6,7]
y=[2,3,1,4,5,3,6]
plt.scatter(x,y)
plt.title('scatter plot')
plt.show()
```



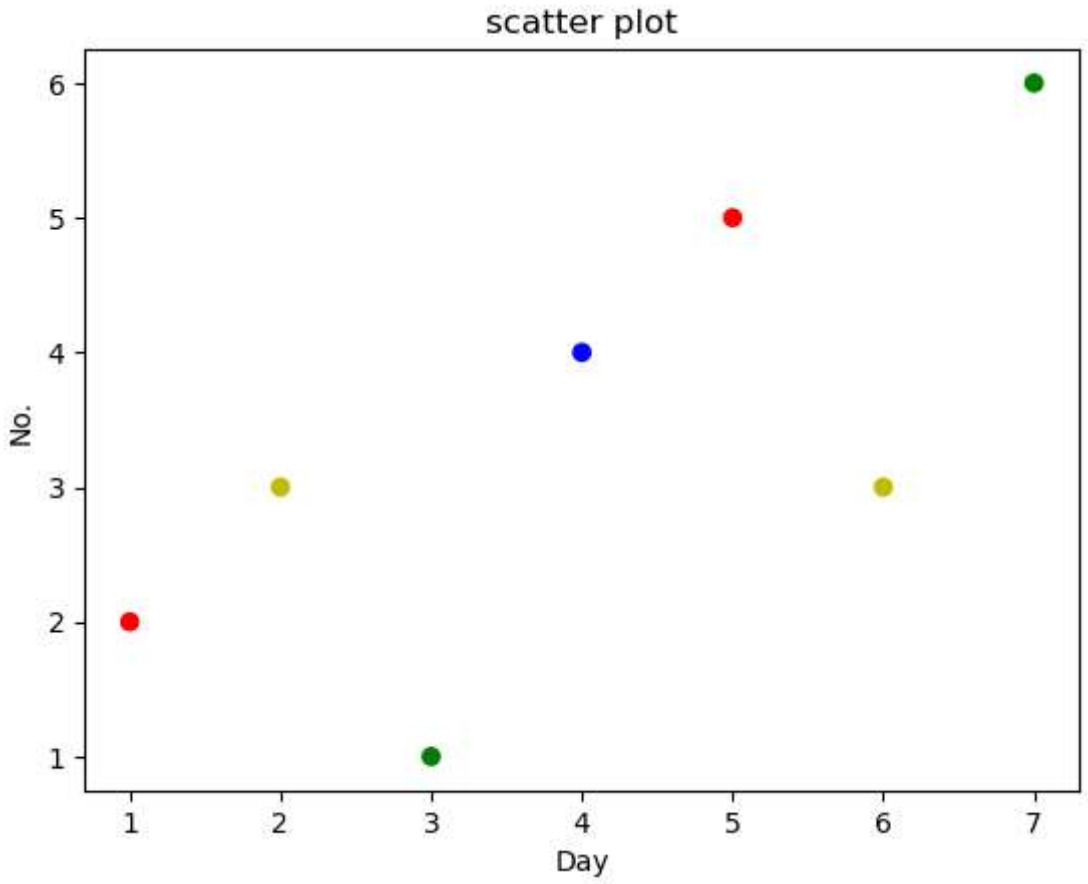
In [25]:
```
Day = [1,2,3,4,5,6,7]
no =[2,3,1,4,5,3,6]
plt.scatter(Day,no)
plt.title('scatter plot')
plt.xlabel("Day")
plt.ylabel("No.")
plt.show()
```



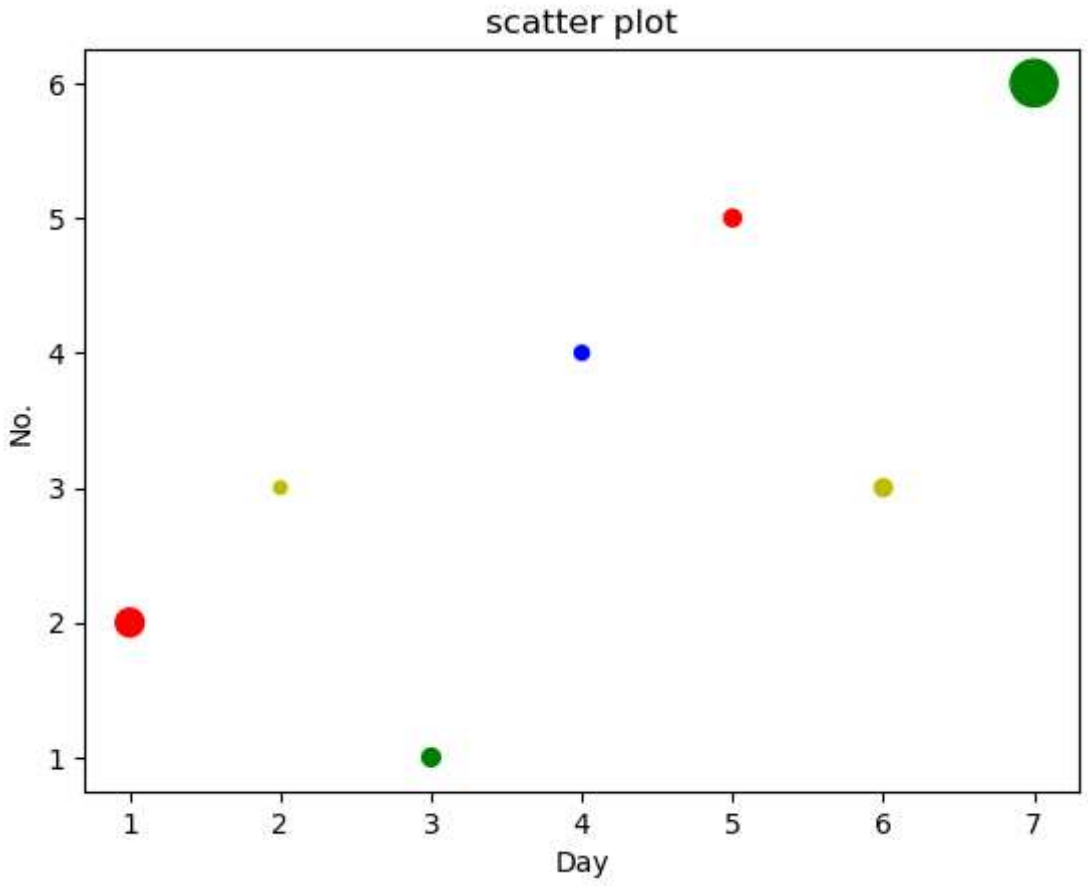
In [26]:
```
Day = [1,2,3,4,5,6,7]
no =[2,3,1,4,5,3,6]
plt.scatter(Day,no,color='r')
plt.title('scatter plot')
plt.xlabel("Day")
plt.ylabel("No.")
plt.show()
```



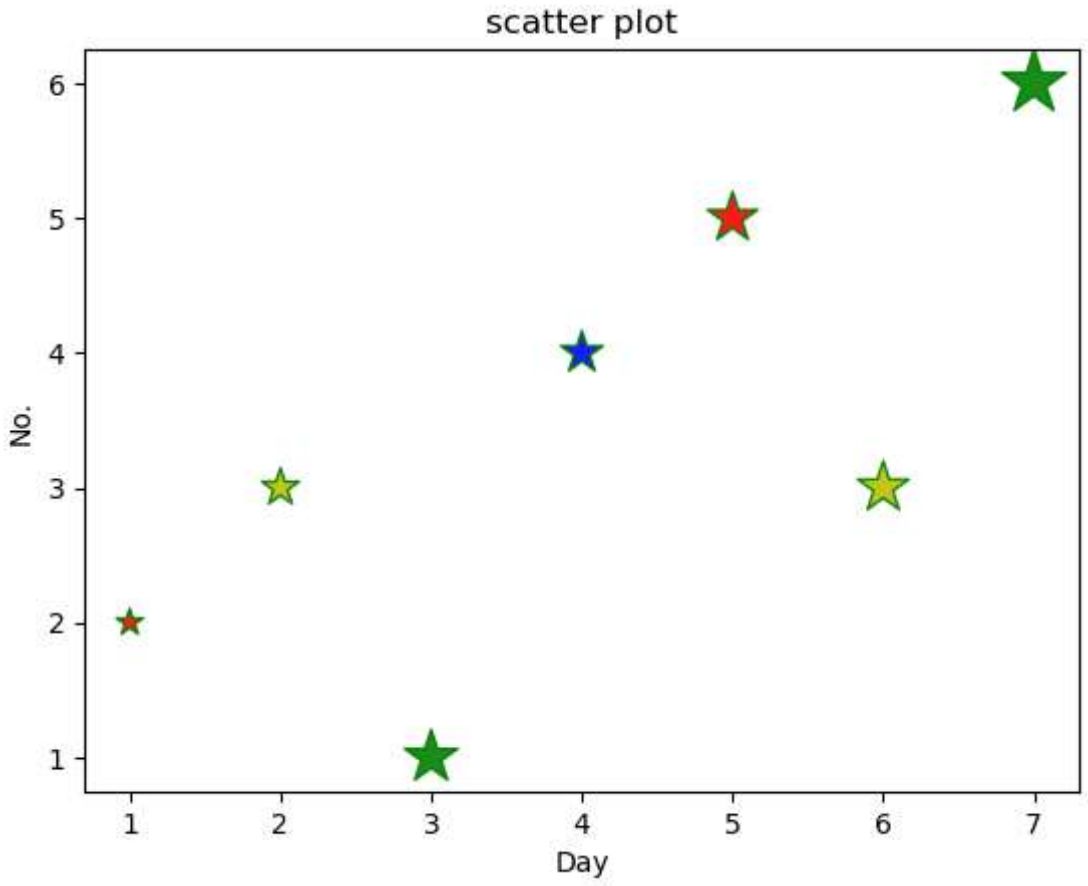
In [37]:
```
Day = [1,2,3,4,5,6,7]
no =[2,3,1,4,5,3,6]
colors=["r","y","g","b","r","y","g"]
plt.scatter(Day,no,c=colors)
plt.title('scatter plot')
plt.xlabel("Day")
plt.ylabel("No.")
plt.show()
```



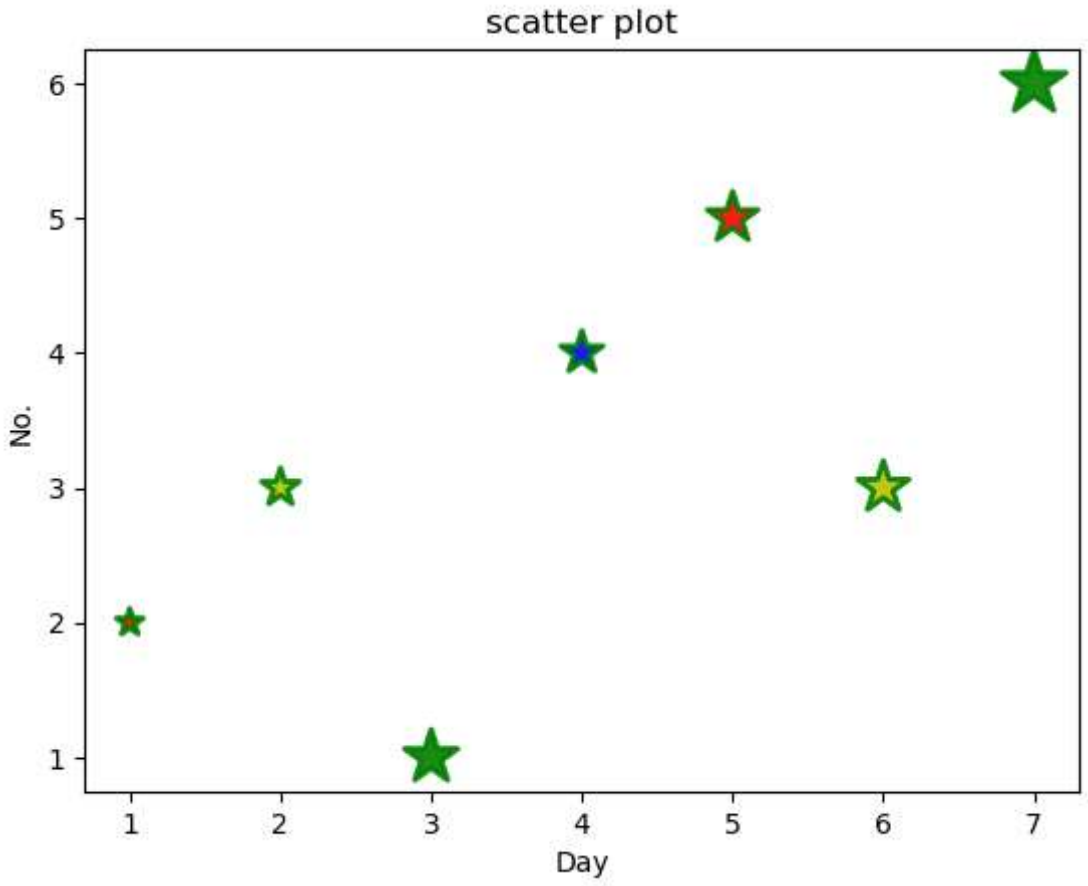
In [40]:
```
Day = [1,2,3,4,5,6,7]
no =[2,3,1,4,5,3,6]
colors=["r","y","g","b","r","y","g"]
sizes=[100,20,40,25,36,37,280]
plt.scatter(Day,no,c=colors,s= sizes)
plt.title('scatter plot')
plt.xlabel("Day")
plt.ylabel("No.")
plt.show()
```



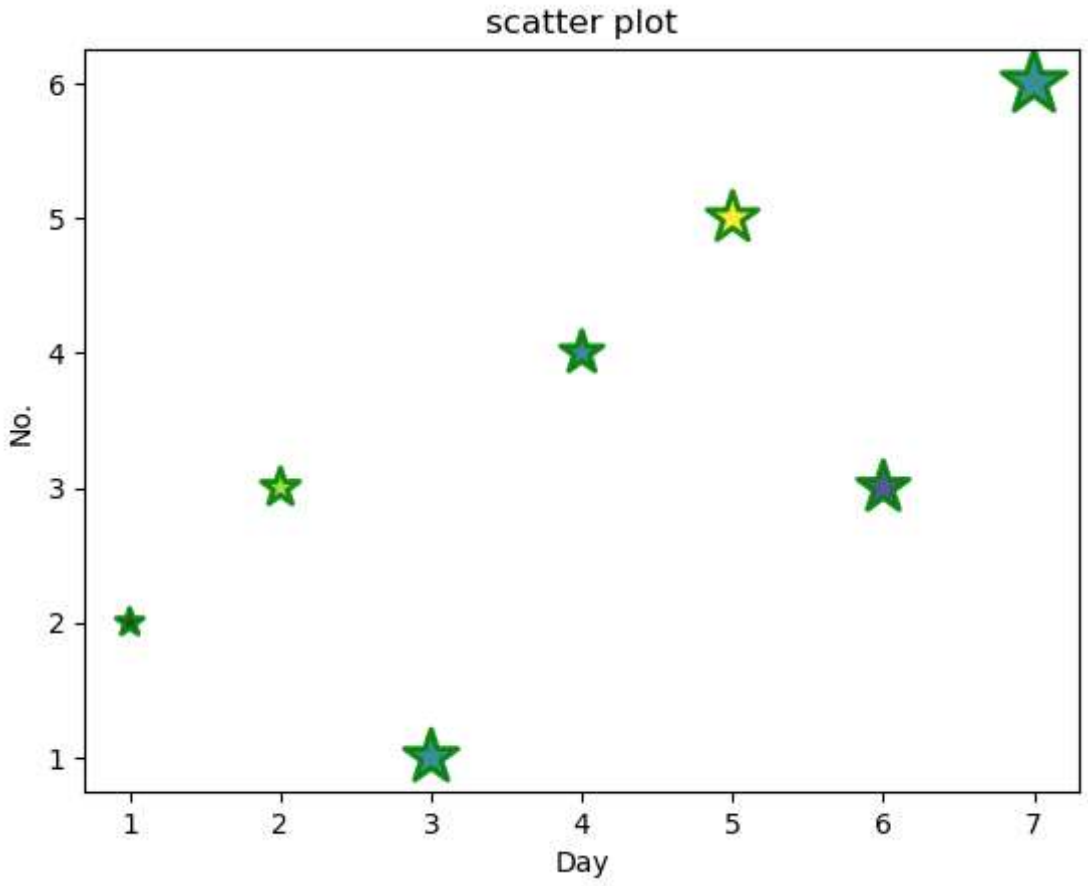
In [48]:
```
Day = [1,2,3,4,5,6,7]
no =[2,3,1,4,5,3,6]
colors=["r","y","g","b","r","y","g"]
sizes=[100,200,400,250,360,370,580]
plt.scatter(Day,no,c=colors,s= sizes,alpha=0.9,marker='*',edgecolor='g')
plt.title('scatter plot')
plt.xlabel("Day")
plt.ylabel("No.")
plt.show()
```



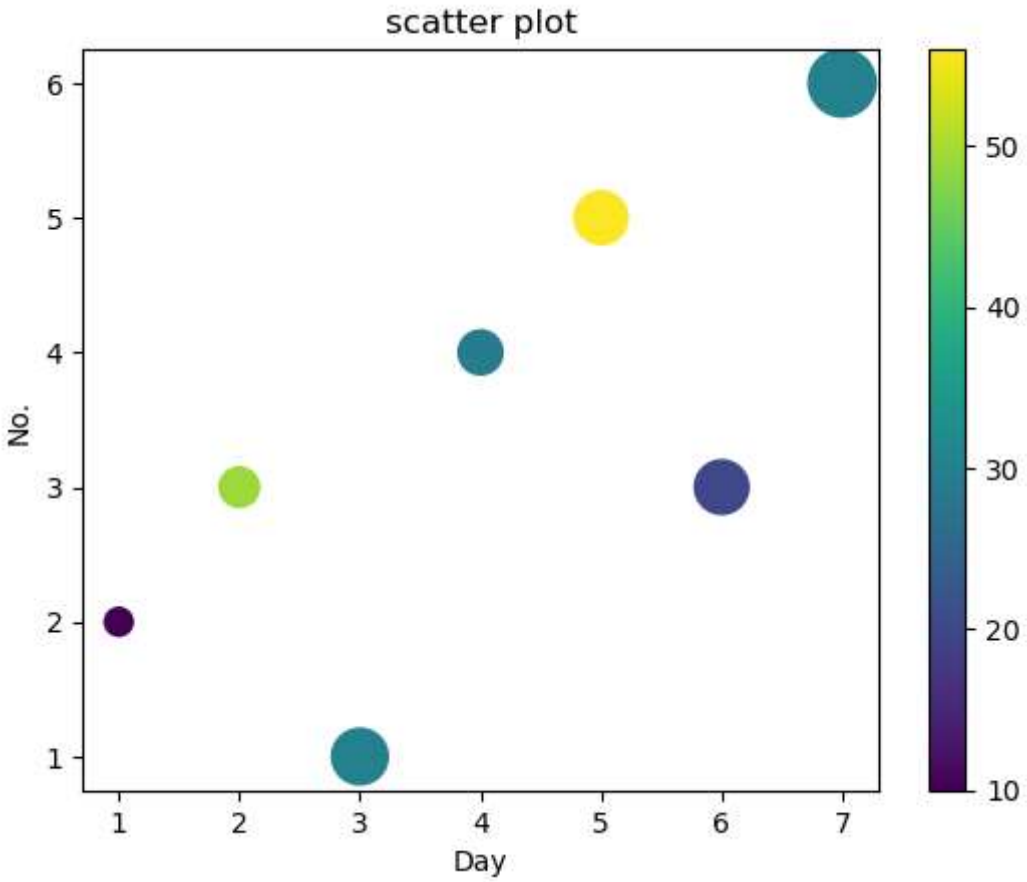
In [49]:
```
Day = [1,2,3,4,5,6,7]
no =[2,3,1,4,5,3,6]
colors=["r","y","g","b","r","y","g"]
sizes=[100,200,400,250,360,370,580]
plt.scatter(Day,no,c=colors,s= sizes,alpha=0.9,marker='*',edgecolor='g',linewi
plt.title('scatter plot')
plt.xlabel("Day")
plt.ylabel("No.")
plt.show()
```



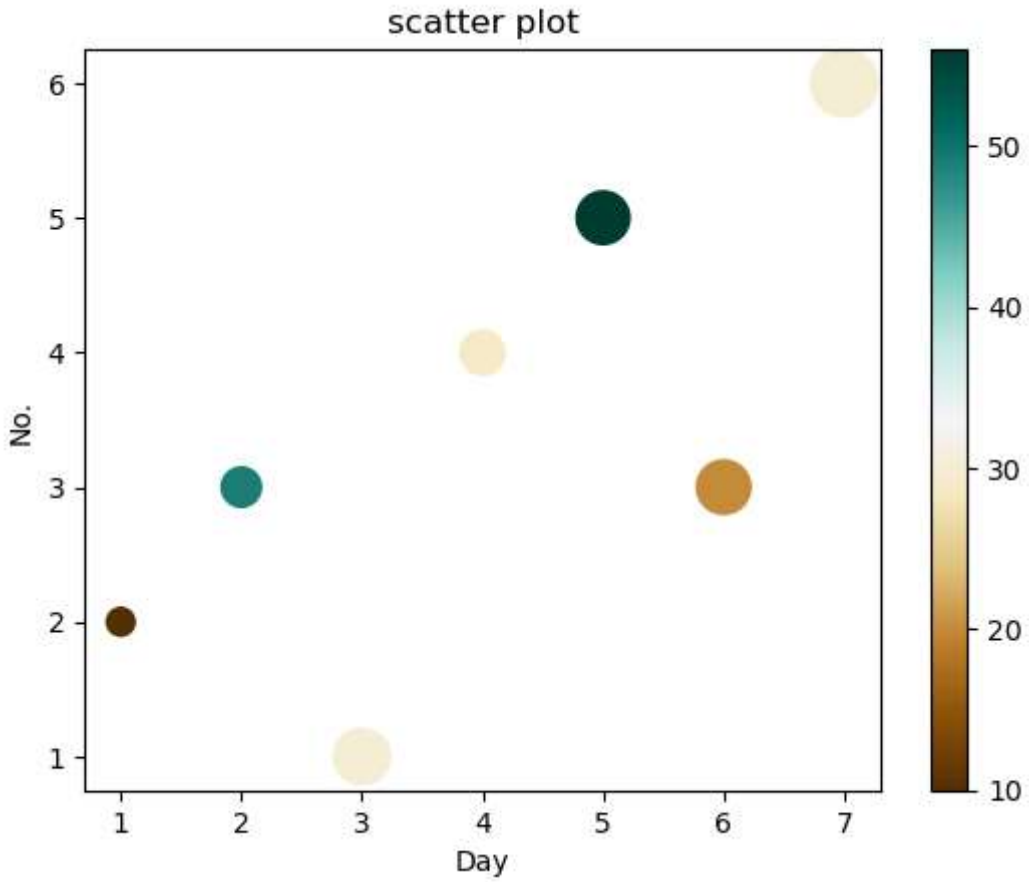
In [50]:
```
Day = [1,2,3,4,5,6,7]
no =[2,3,1,4,5,3,6]
colors=[10,49,30,29,56,20,30]
sizes=[100,200,400,250,360,370,580]
plt.scatter(Day,no,c=colors,s= sizes,alpha=0.9,marker='*',edgecolor='g',linewi
plt.title('scatter plot')
plt.xlabel("Day")
plt.ylabel("No.")
plt.show()
```



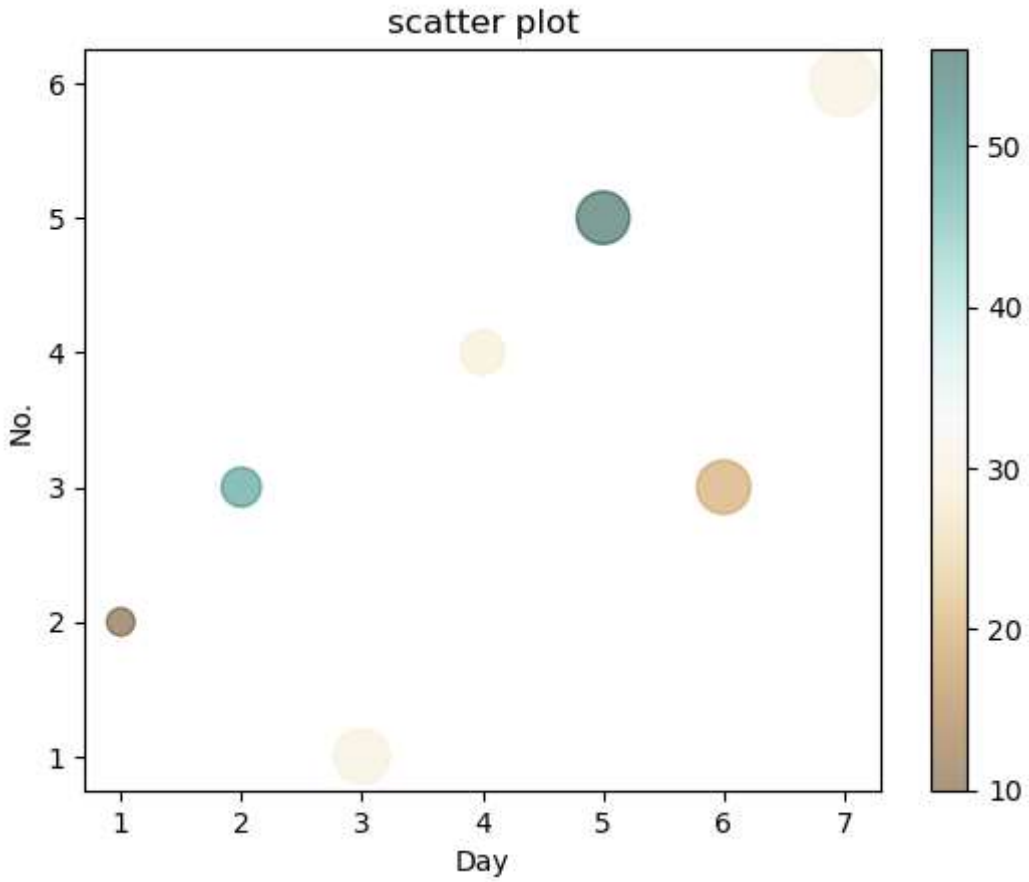
In [52]:
```
Day = [1,2,3,4,5,6,7]
no =[2,3,1,4,5,3,6]
colors=[10,49,30,29,56,20,30]
sizes=[100,200,400,250,360,370,580]
plt.scatter(Day,no,c=colors,s= sizes,cmap="viridis")
plt.colorbar()
plt.title('scatter plot')
plt.xlabel("Day")
plt.ylabel("No.")
plt.show()
```



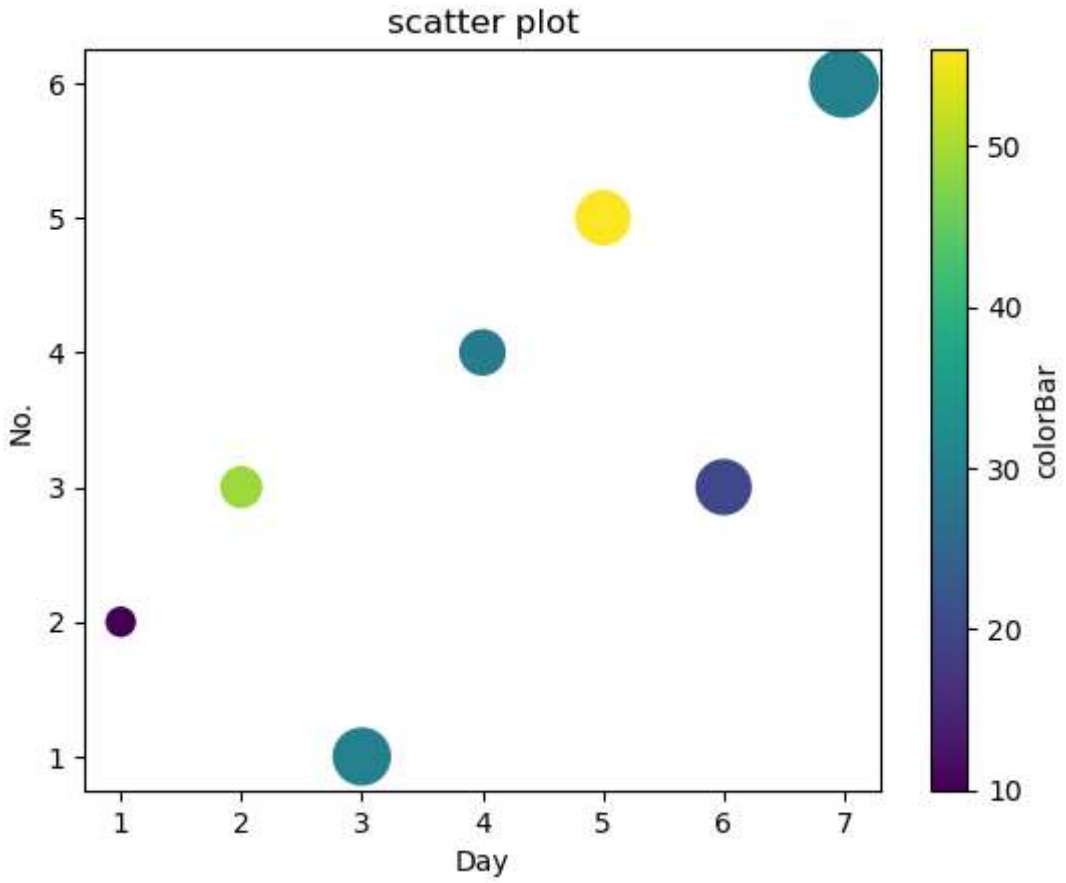
In [54]:
```
Day = [1,2,3,4,5,6,7]
no =[2,3,1,4,5,3,6]
colors=[10,49,30,29,56,20,30]
sizes=[100,200,400,250,360,370,580]
plt.scatter(Day,no,c=colors,s= sizes,cmap="BrBG")
plt.colorbar()
plt.title('scatter plot')
plt.xlabel("Day")
plt.ylabel("No.")
plt.show()
```



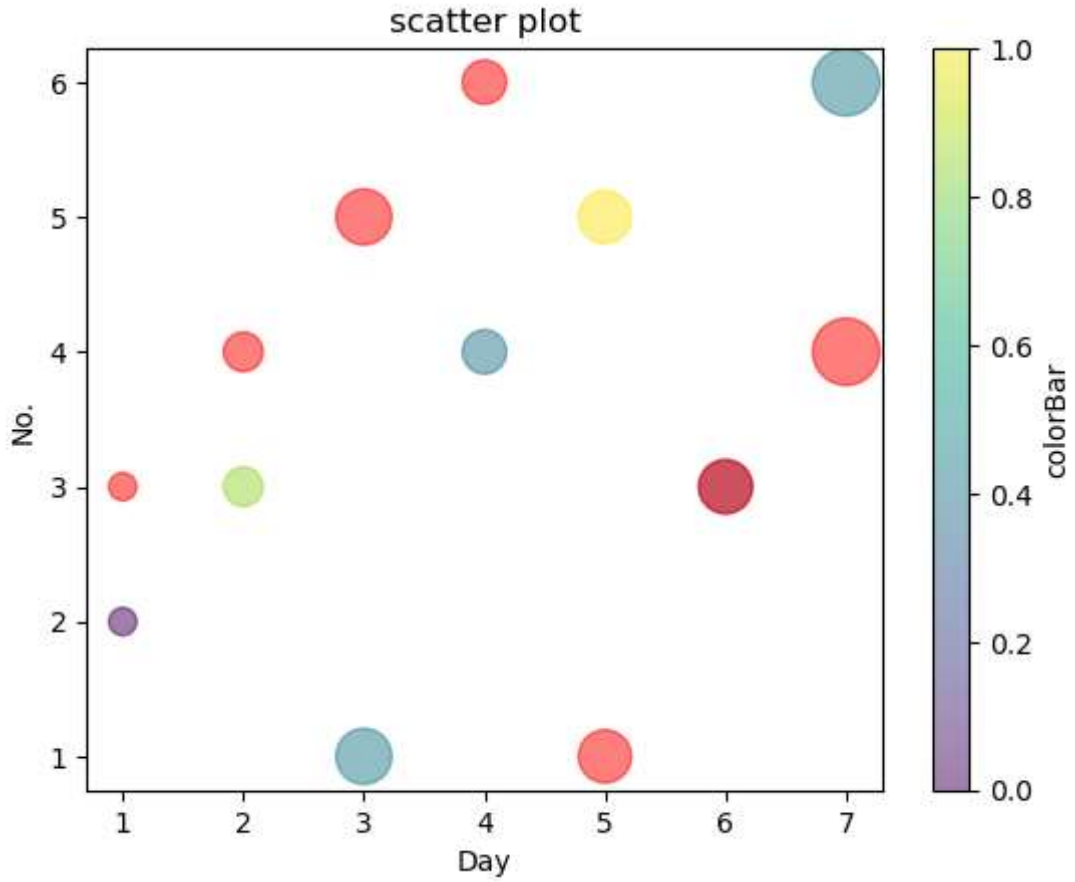
In [55]:
```
Day = [1,2,3,4,5,6,7]
no =[2,3,1,4,5,3,6]
colors=[10,49,30,29,56,20,30]
sizes=[100,200,400,250,360,370,580]
plt.scatter(Day,no,c=colors,s= sizes,cmap="BrBG",alpha=0.5)
plt.colorbar()
plt.title('scatter plot')
plt.xlabel("Day")
plt.ylabel("No.")
plt.show()
```



In [56]:
```
Day = [1,2,3,4,5,6,7]
no =[2,3,1,4,5,3,6]
colors=[10,49,30,29,56,20,30]
sizes=[100,200,400,250,360,370,580]
plt.scatter(Day,no,c=colors,s= sizes,cmap="viridis")
t=plt.colorbar()
t.set_label("colorBar")
plt.title('scatter plot')
plt.xlabel("Day")
plt.ylabel("No.")
plt.show()
```

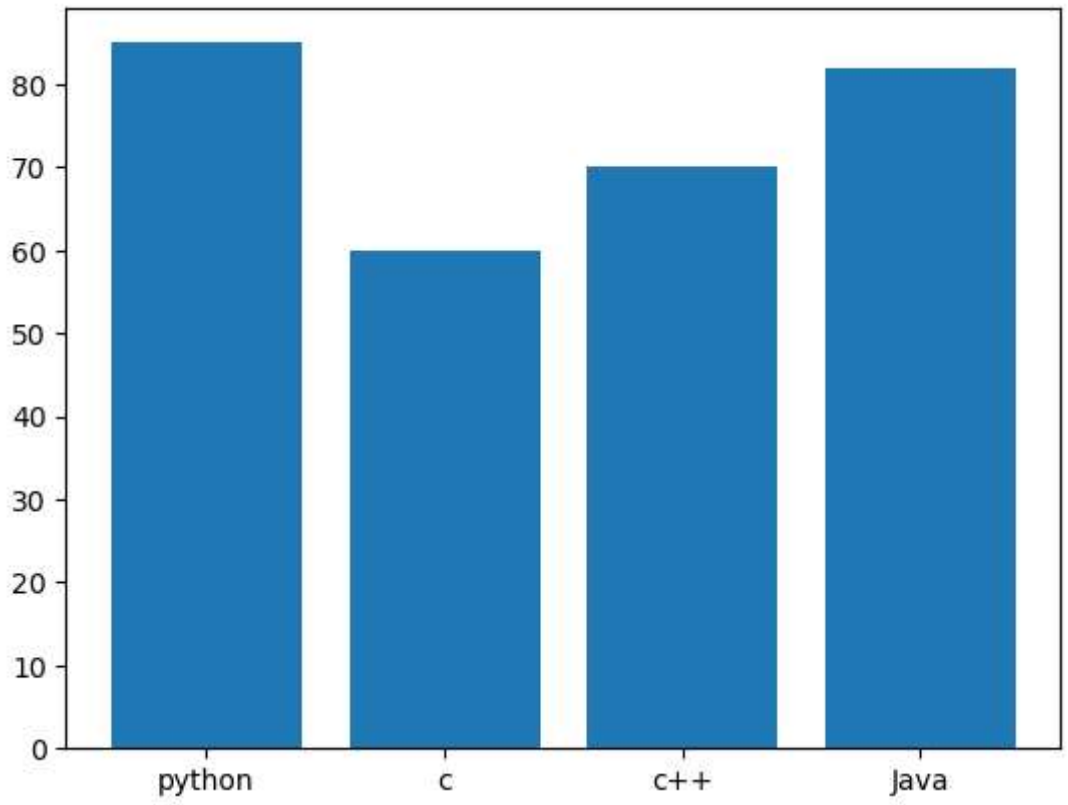


In [58]:
```
Day = [1,2,3,4,5,6,7]
no1 =[2,3,1,4,5,3,6]
no2 =[3,4,5,6,1,3,4]
colors=[10,49,30,29,56,20,30]
sizes=[100,200,400,250,360,370,580]
plt.scatter(Day,no1,c=colors,s= sizes,cmap="viridis",alpha=0.5)
plt.scatter(Day,no2,c='r',s= sizes,alpha=0.5)
t=plt.colorbar()
t.set_label("colorBar")
plt.title('scatter plot')
plt.xlabel("Day")
plt.ylabel("No.")
plt.show()
```

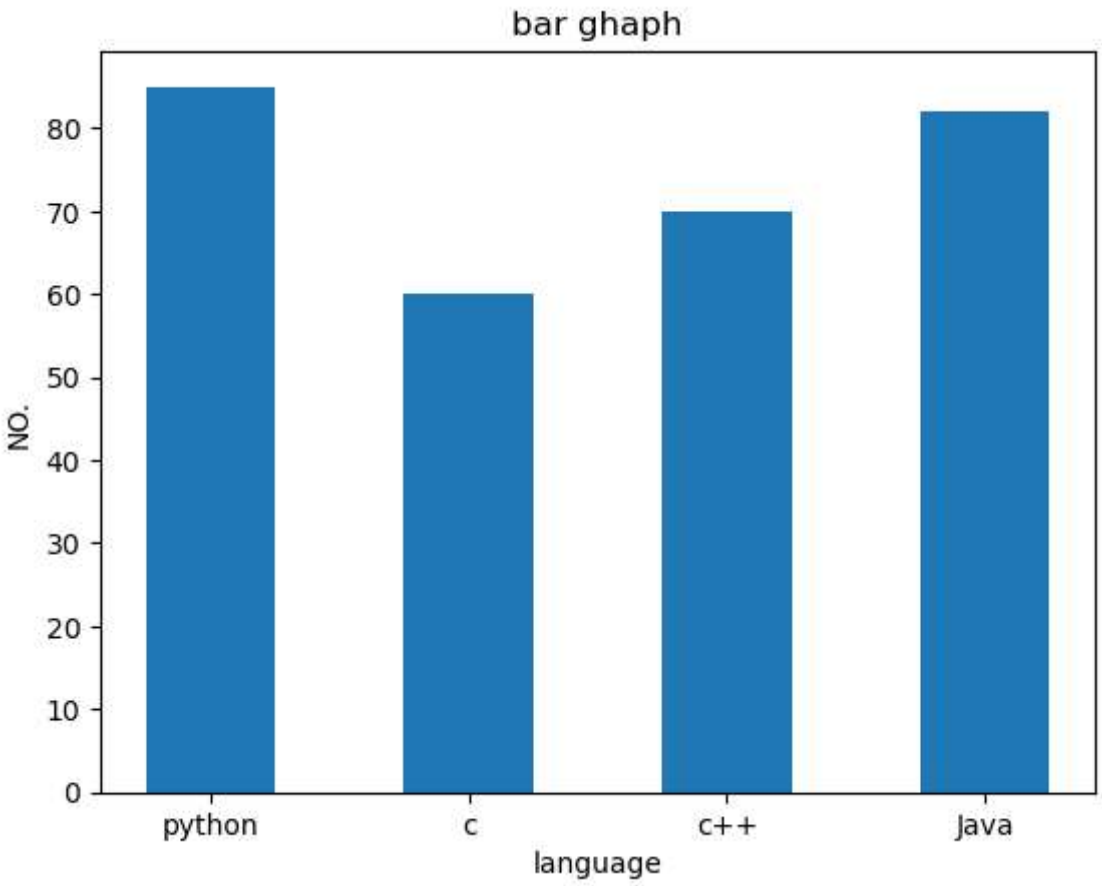


Bar plot

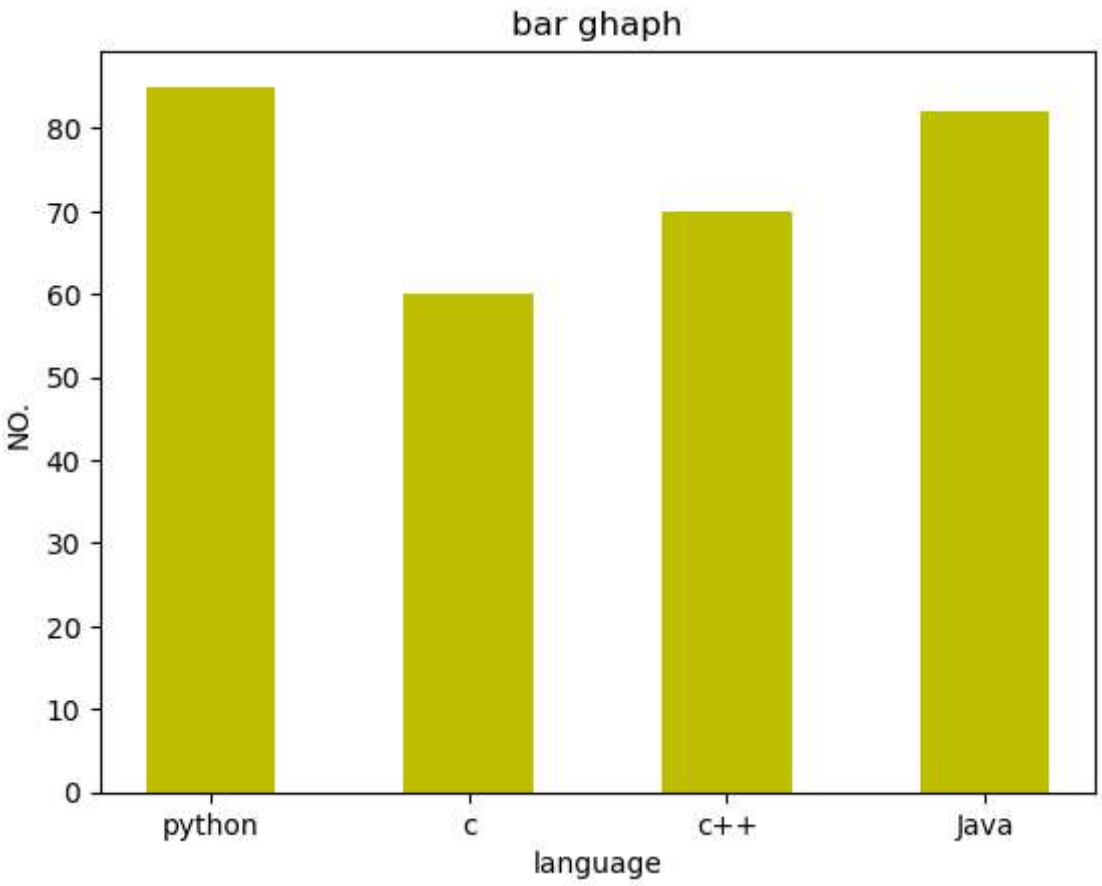
In [59]:
```
x = ['python','c','c++','Java']
y = [85,60,70,82]
plt.bar(x,y)
plt.show()
```



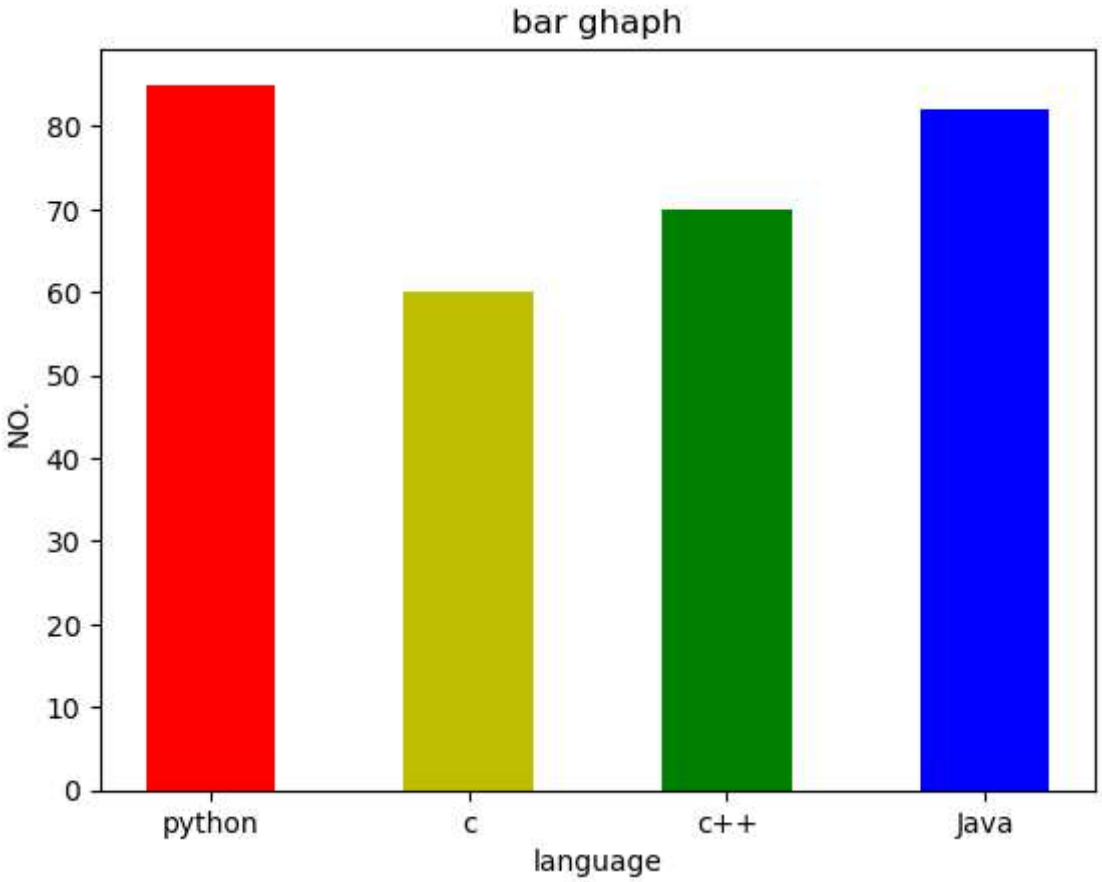
In [61]:
```
x = ['python','c','c++','Java']
y = [85,60,70,82]
plt.xlabel("language")
plt.ylabel("NO.")
plt.title("bar ghaph")
plt.bar(x,y,width= 0.5)
plt.show()
```



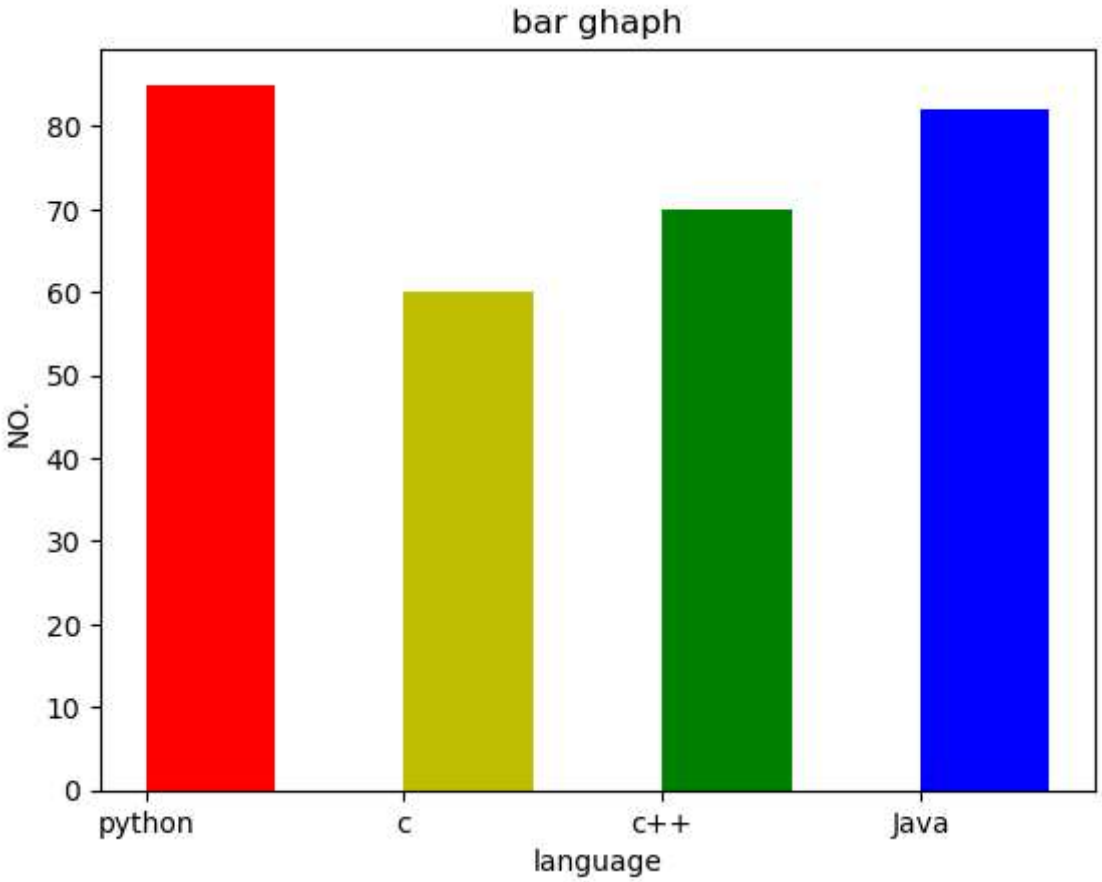
In [62]:
```
x = ['python','c','c++','Java']
y = [85,60,70,82]
plt.xlabel("language")
plt.ylabel("NO.")
plt.title("bar ghaph")
plt.bar(x,y,width= 0.5,color='y')
plt.show()
```



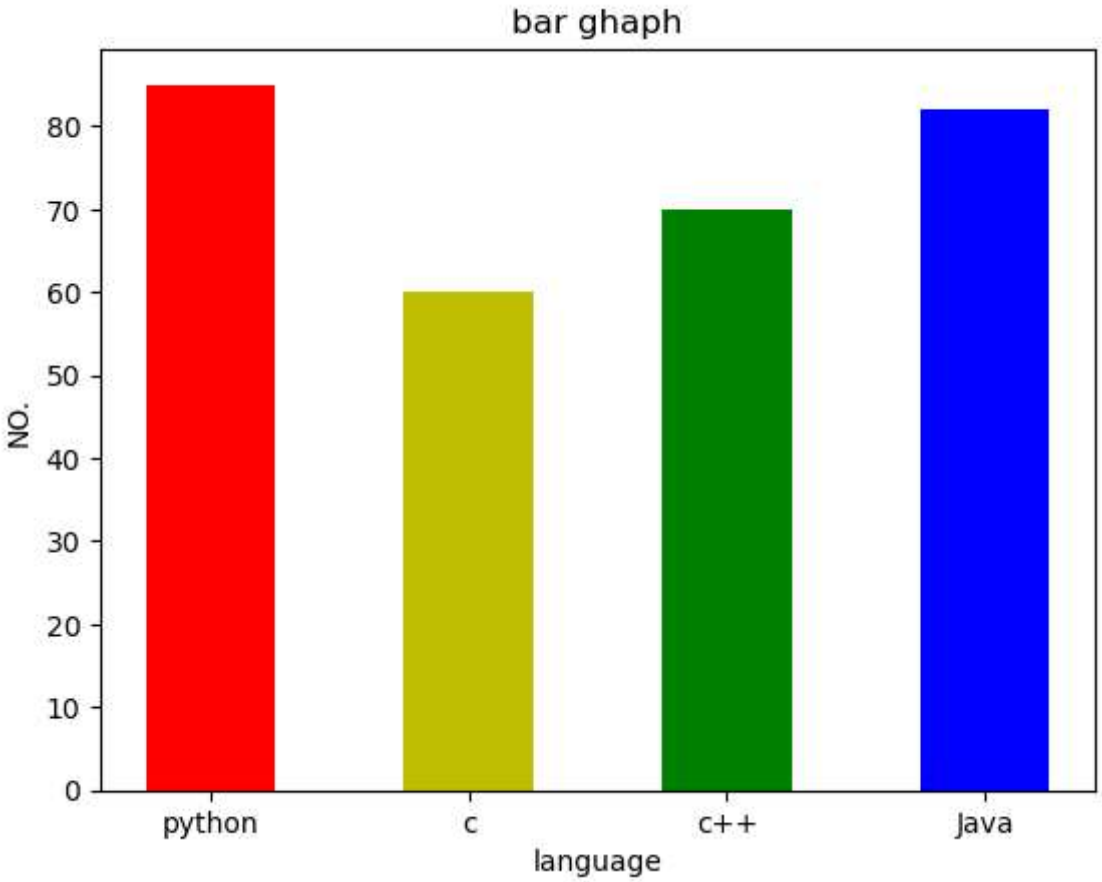
In [63]:
```
x = ['python','c','c++','Java']
y = [85,60,70,82]
plt.xlabel("language")
plt.ylabel("NO.")
plt.title("bar ghaph")
c=['r','y','g','b']
plt.bar(x,y,width= 0.5,color=c)
plt.show()
```



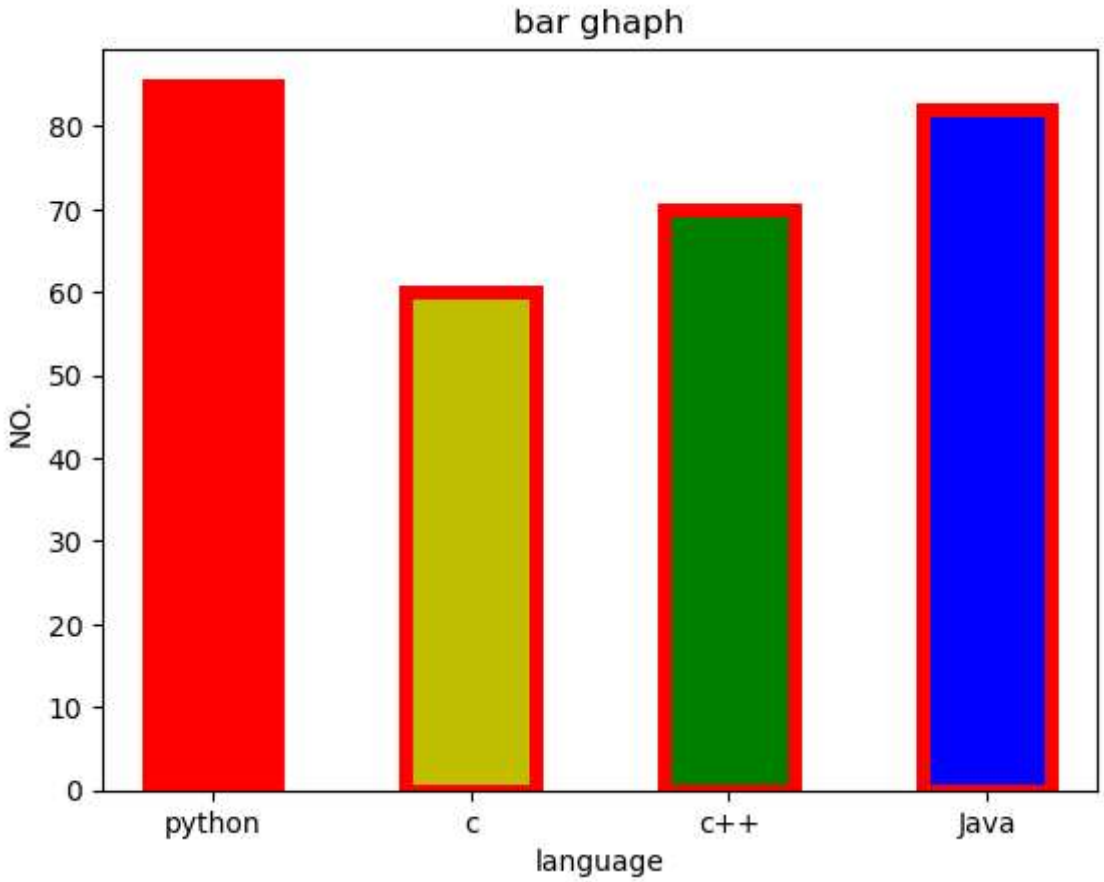
In [64]:
```
x = ['python','c','c++','Java']
y = [85,60,70,82]
plt.xlabel("language")
plt.ylabel("NO.")
plt.title("bar ghaph")
c=['r','y','g','b']
plt.bar(x,y,width= 0.5,color=c,align='edge')
plt.show()
```



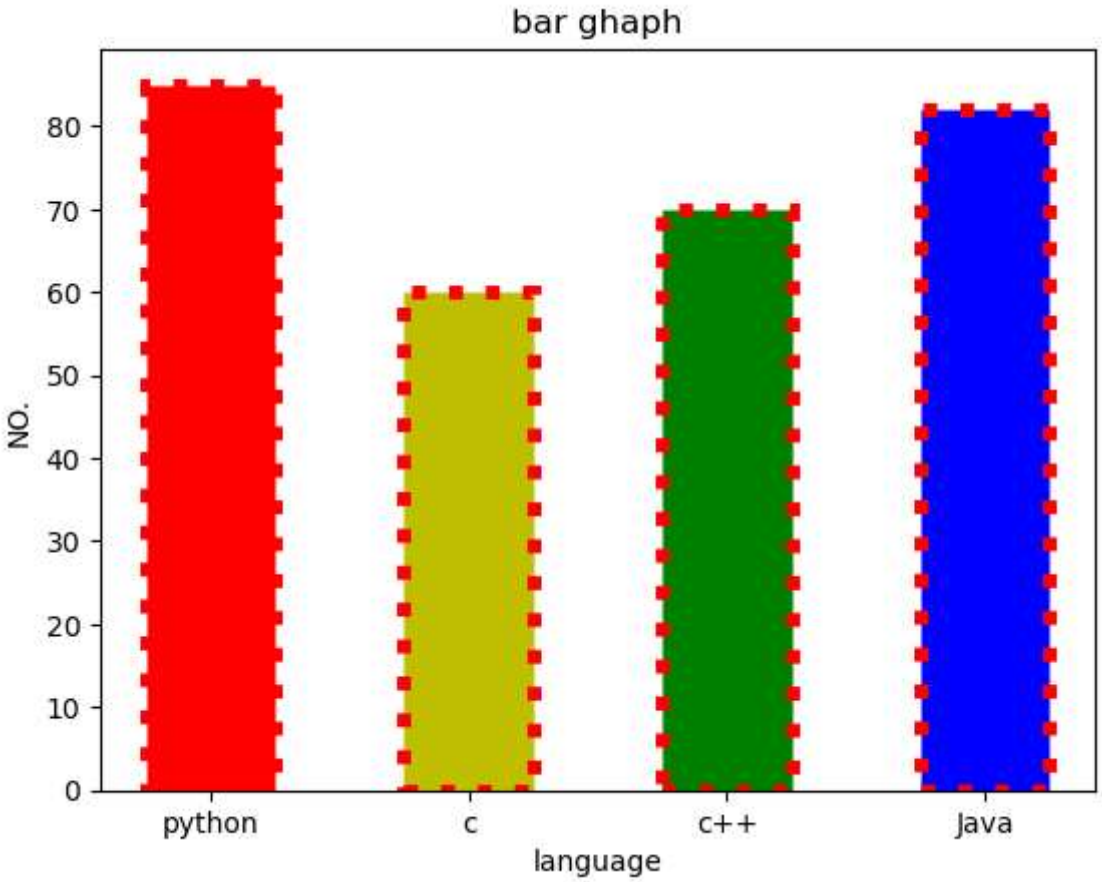
In [65]:
```
x = ['python','c','c++','Java']
y = [85,60,70,82]
plt.xlabel("language")
plt.ylabel("NO.")
plt.title("bar ghaph")
c=['r','y','g','b']
plt.bar(x,y,width= 0.5,color=c,align='center')
plt.show()
```



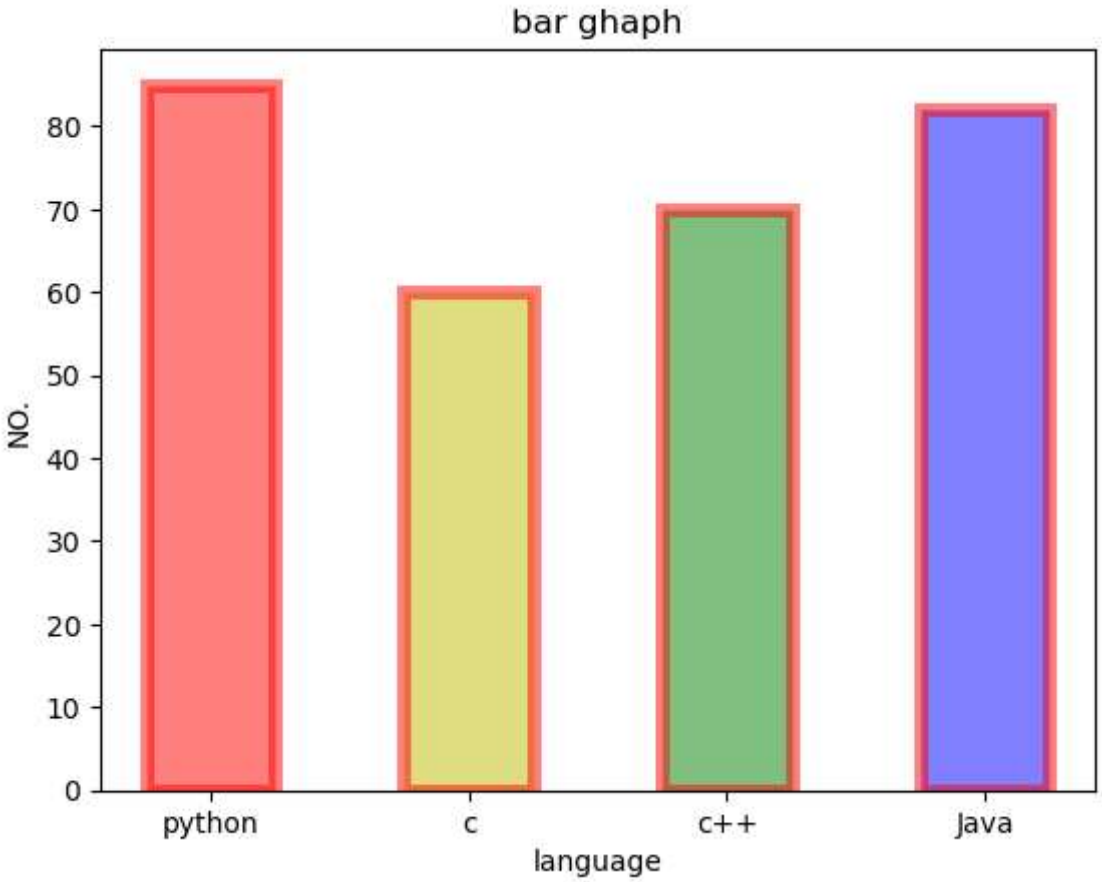
In [67]:
```
x = ['python','c','c++','Java']
y = [85,60,70,82]
plt.xlabel("language")
plt.ylabel("NO.")
plt.title("bar ghaph")
c=['r','y','g','b']
plt.bar(x,y,width= 0.5,color=c,edgecolor='r',linewidth=5)
plt.show()
```



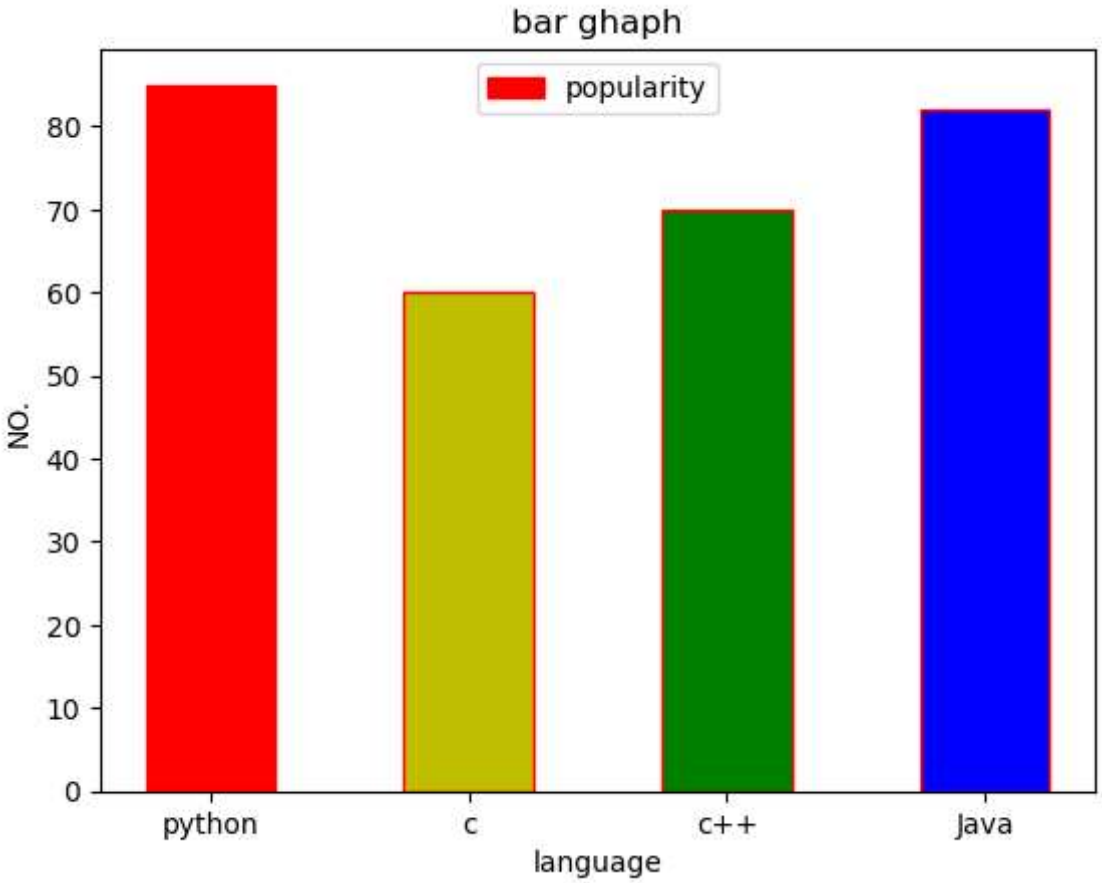
In [68]:
```
x = ['python','c','c++','Java']
y = [85,60,70,82]
plt.xlabel("language")
plt.ylabel("NO.")
plt.title("bar ghaph")
c=['r','y','g','b']
plt.bar(x,y,width= 0.5,color=c,edgecolor='r',linewidth=5,linestyle=':')
plt.show()
```



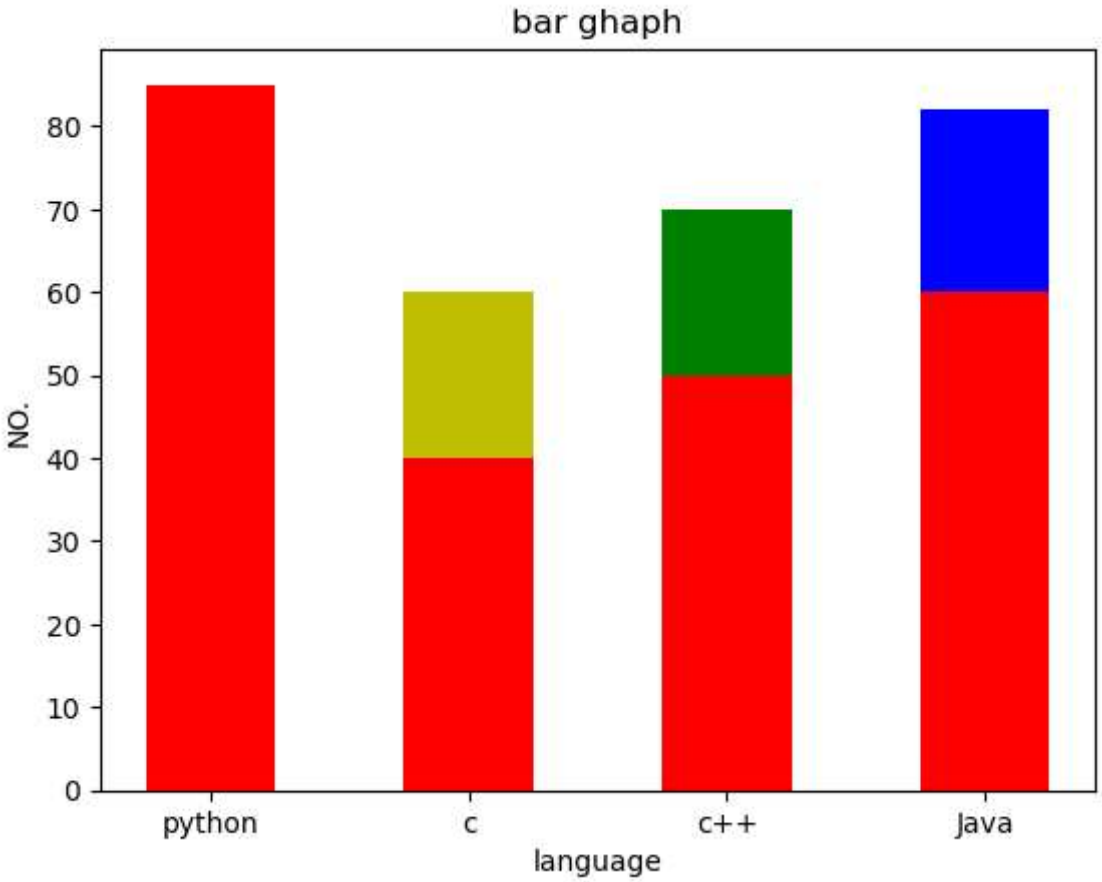
In [69]:
```
x = ['python','c','c++','Java']
y = [85,60,70,82]
plt.xlabel("language")
plt.ylabel("NO.")
plt.title("bar ghaph")
c=['r','y','g','b']
plt.bar(x,y,width= 0.5,color=c,edgecolor='r',linewidth=5,alpha=0.5)
plt.show()
```



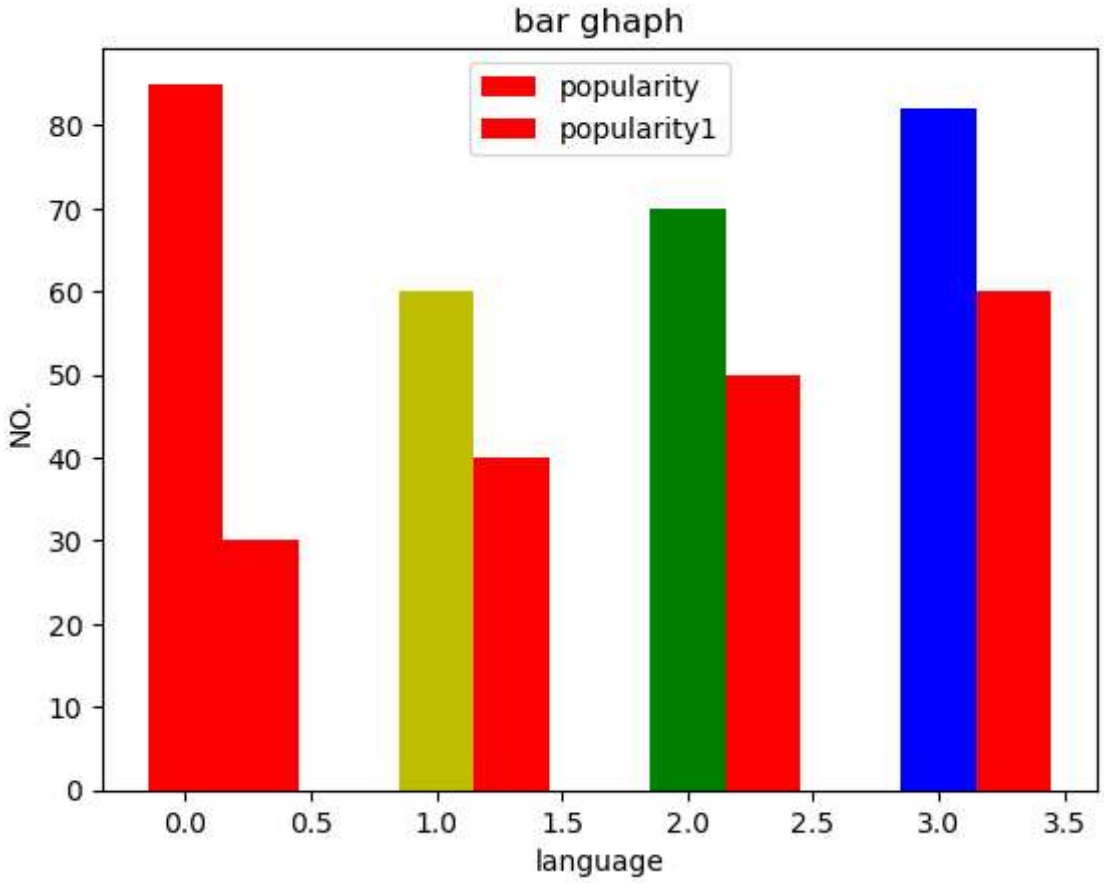
In [72]:
```
x = ['python','c','c++','Java']
y = [85,60,70,82]
plt.xlabel("language")
plt.ylabel("NO.")
plt.title("bar ghaph")
c=['r','y','g','b']
plt.bar(x,y,width= 0.5,color=c,edgecolor='r',label='popularity')
plt.legend()
plt.show()
```



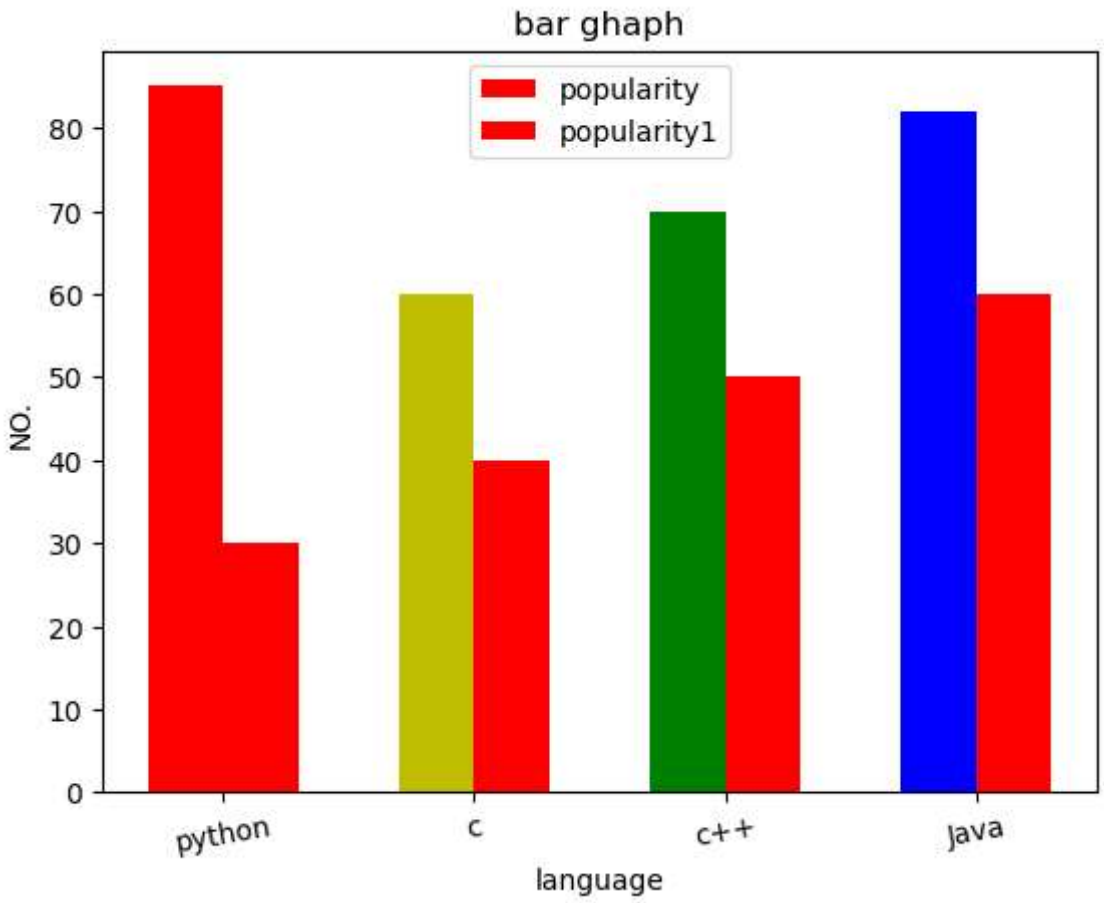
In [74]:
```
x = ['python','c','c++','Java']
y = [85,60,70,82]
z=[30,40,50,60]
plt.xlabel("language")
plt.ylabel("NO.")
plt.title("bar ghaph")
c=['r','y','g','b']
plt.bar(x,y,width= 0.5,color=c,label='popularity')
plt.bar(x,z,width= 0.5,color="r",label='popularity1')
plt.show()
```



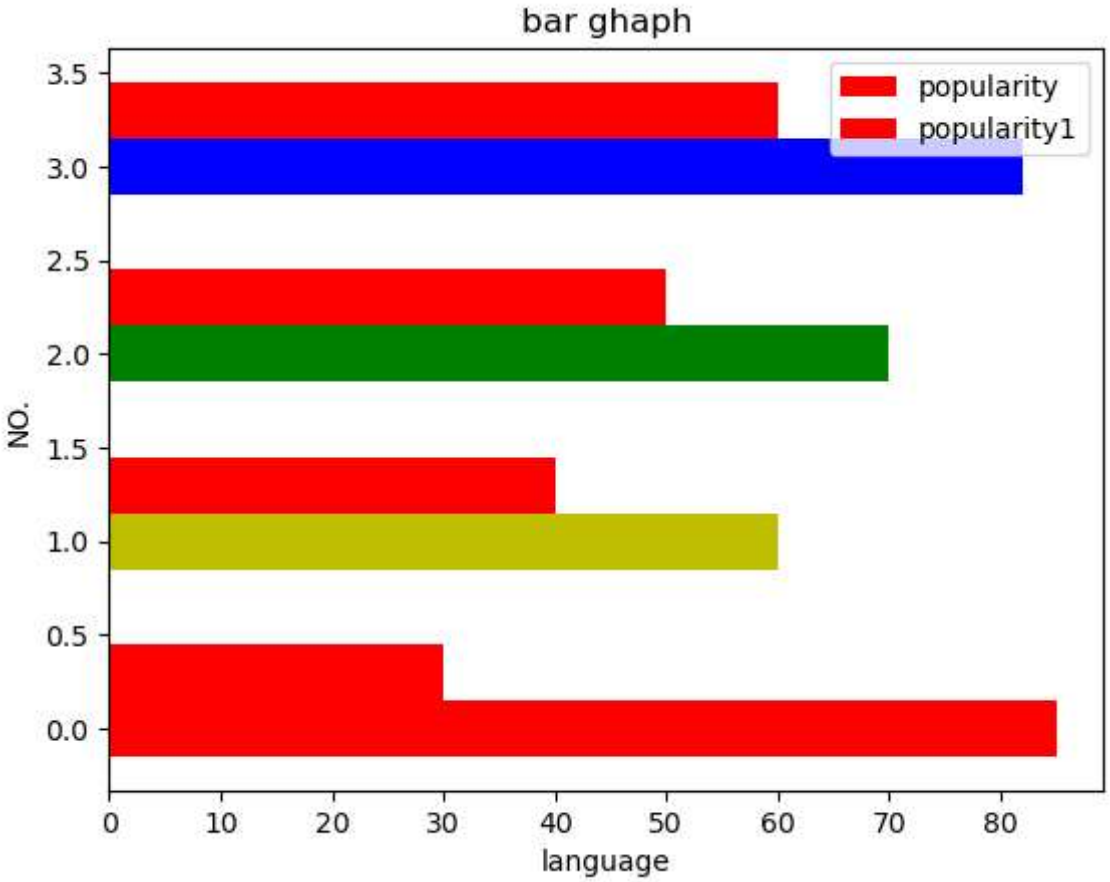
In [82]:
```
import numpy as np
x = ['python','c','c++','Java']
y = [85,60,70,82]
z=[30,40,50,60]
p=np.arange(len(x))
p1=[j+width for j in p]
width=0.3
plt.xlabel("language")
plt.ylabel("NO.")
plt.title("bar ghaph")
c=['r','y','g','b']

plt.bar(p,y,width,color=c,label="popularity")
plt.bar(p1,z,width,color="r",label="popularity1")
plt.legend()
plt.show()
```



In [85]:
```
x = ['python','c','c++','Java']
y = [85,60,70,82]
z=[30,40,50,60]
p=np.arange(len(x))
p1=[j+width for j in p]
width=0.3
plt.xlabel("language")
plt.ylabel("NO.")
plt.title("bar ghaph")
c=['r','y','g','b']

plt.bar(p,y,width,color=c,label="popularity")
plt.bar(p1,z,width,color="r",label="popularity1")
plt.xticks(p+width/2,x,rotation=10)
plt.legend()

plt.show()
```



In [87]:
```
x = ['python','c','c++','Java']
y = [85,60,70,82]
z=[30,40,50,60]
p=np.arange(len(x))
p1=[j+width for j in p]
width=0.3
plt.xlabel("language")
plt.ylabel("NO.")
plt.title("bar ghaph")
c=['r','y','g','b']

plt.barh(p,y,width,color=c,label="popularity")
plt.barh(p1,z,width,color="r",label="popularity1")

plt.legend()

plt.show()
```



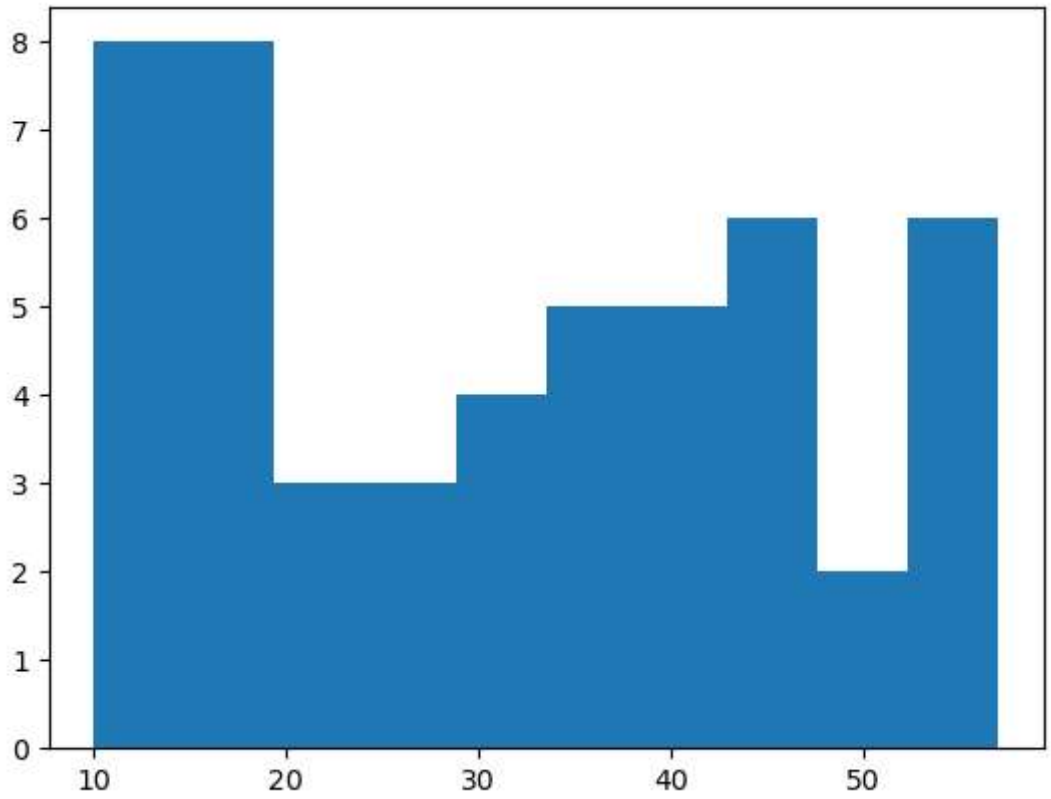
Histrogram

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

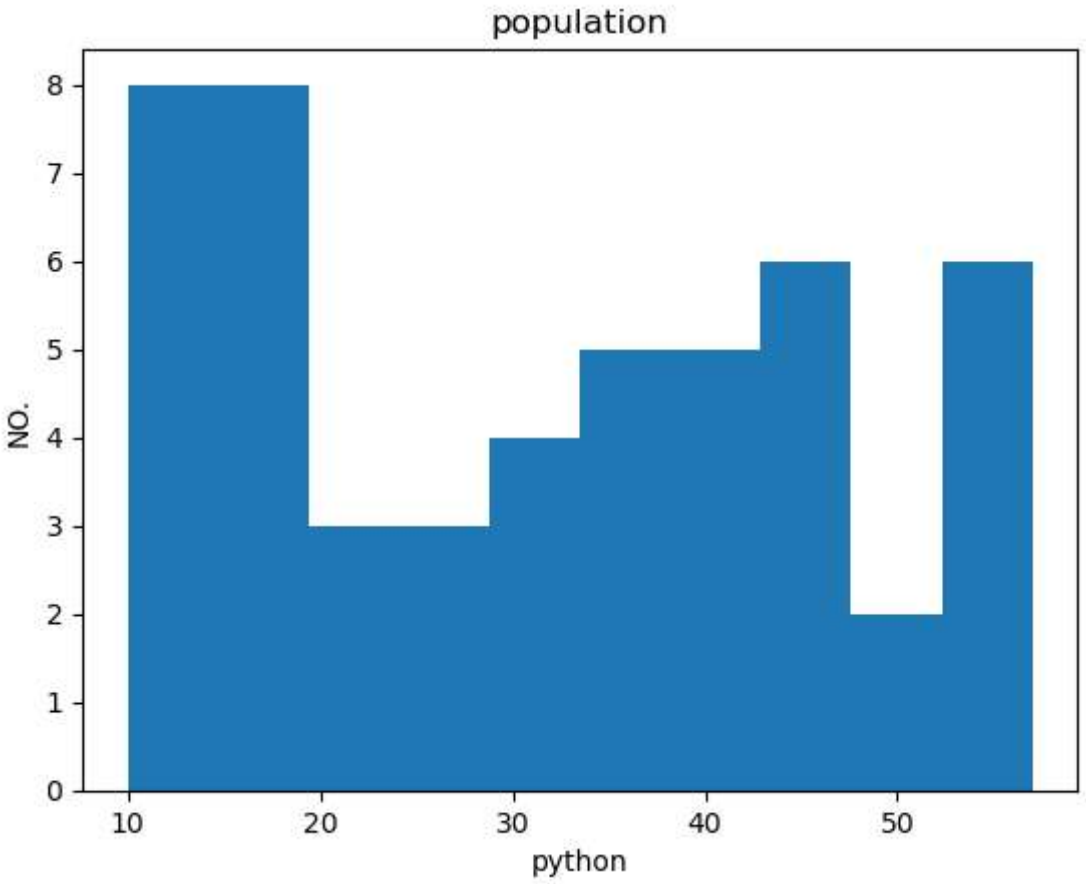
In [2]:
```
x = np.random.randint(10,60,(50))
print(x)
```

```
[11 30 40 43 33 14 47 57 10 34 47 56 36 32 37 14 45 35 16 19 50 17 40 33
 26 54 10 40 43 16 18 54 56 25 28 13 18 17 19 42 45 22 14 51 36 54 42 10
 21 21]
```

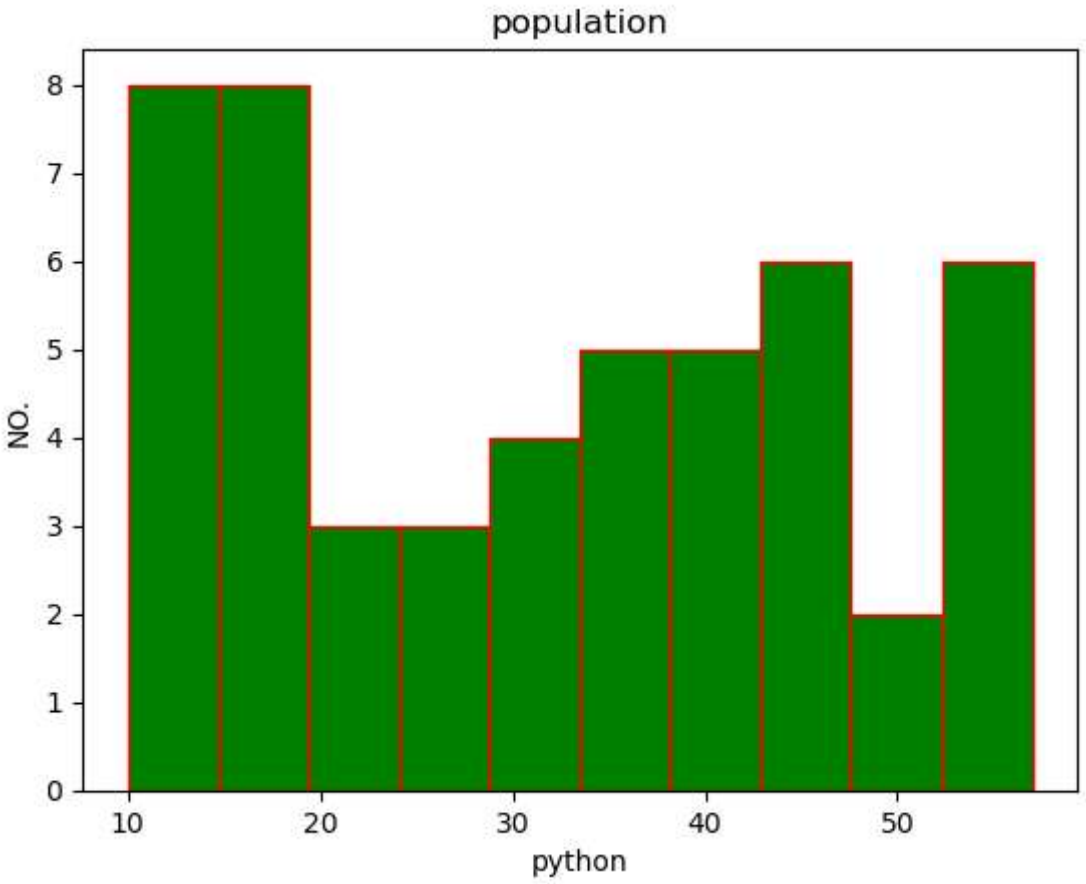
In [4]:
```
no = [11, 30, 40, 43, 33 ,14 ,47, 57, 10 ,34 ,47 ,56, 36, 32 ,37 ,14 ,45, 35, 1
 26, 54, 10, 40, 43, 16, 18, 54, 56, 25, 28, 13, 18, 17, 19, 42, 45, 22 ,14, 51
 21, 21]
plt.hist(no)
plt.show()
```



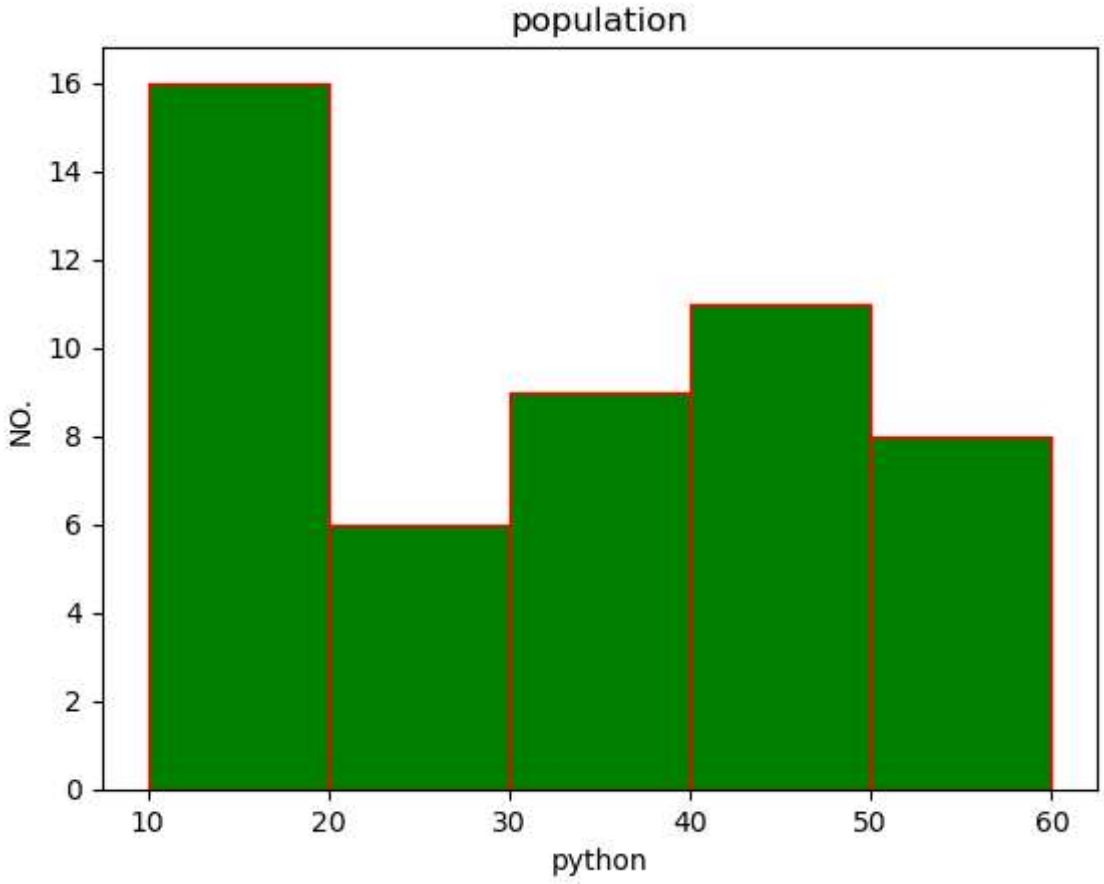
In [5]:
```
no = [11, 30, 40, 43, 33 ,14 ,47, 57, 10 ,34 ,47 ,56, 36, 32 ,37 ,14 ,45, 35, 1
 26, 54, 10, 40, 43, 16, 18, 54, 56, 25, 28, 13, 18, 17, 19, 42, 45, 22 ,14, 5
 21, 21]
plt.hist(no)
plt.title('population')
plt.xlabel("python")
plt.ylabel("NO.")
plt.show()
```



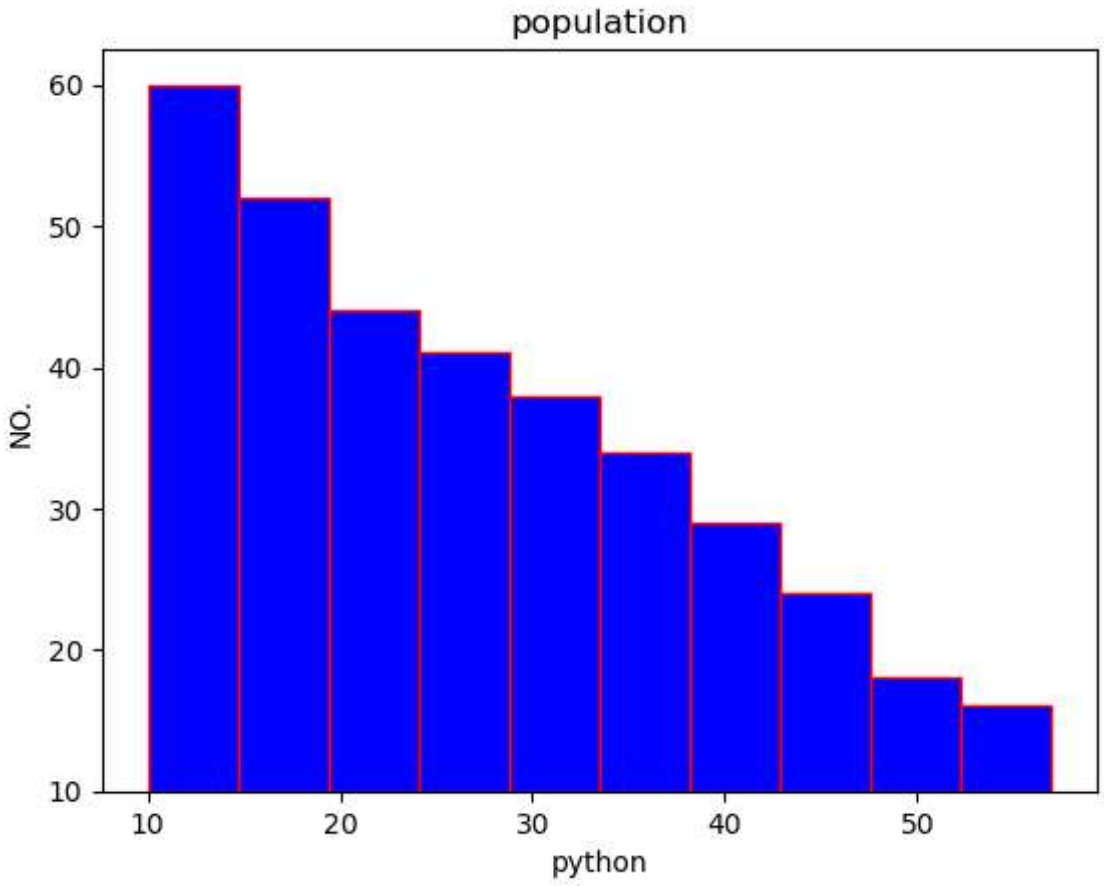
In [8]:
```
no = [11, 30, 40, 43, 33 ,14 ,47, 57, 10 ,34 ,47 ,56, 36, 32 ,37 ,14 ,45, 35, 1
 26, 54, 10, 40, 43, 16, 18, 54, 56, 25, 28, 13, 18, 17, 19, 42, 45, 22 ,14, 51
 21, 21]
plt.hist(no,color='g',edgecolor= "r")
plt.title('population')
plt.xlabel("python")
plt.ylabel("NO.")
plt.show()
```



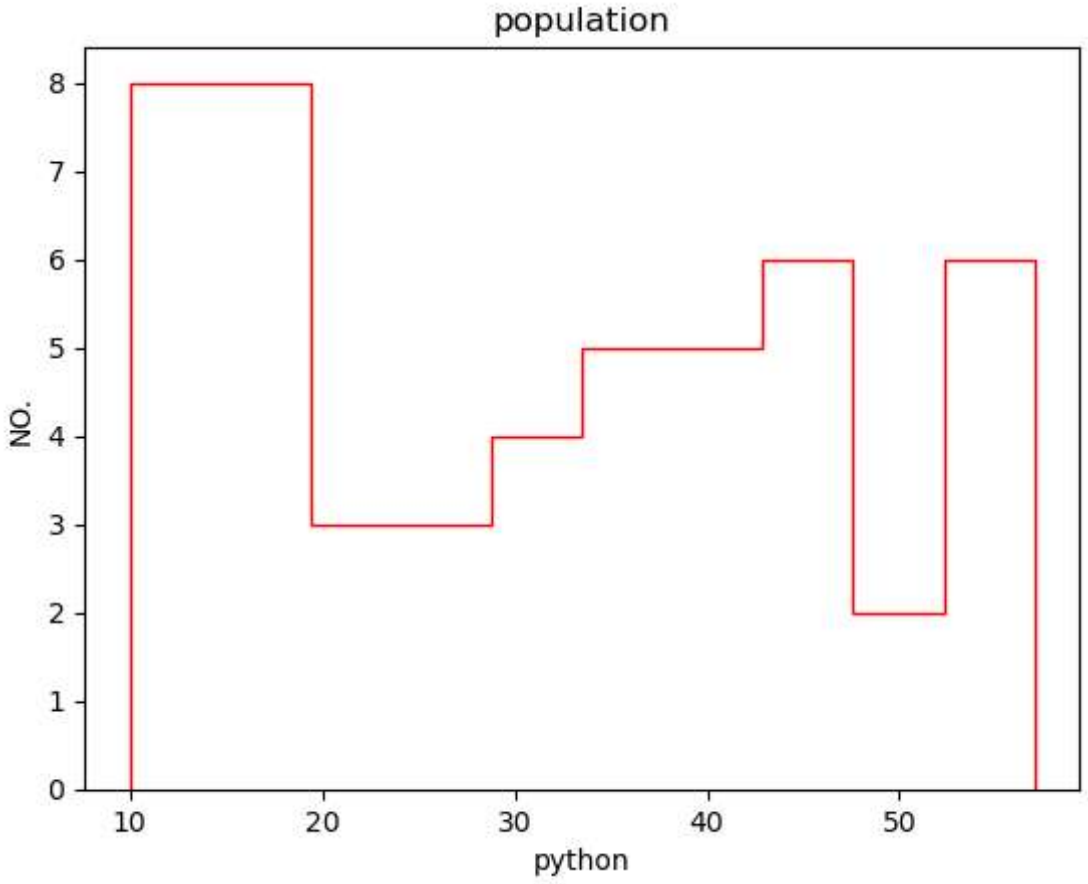
In [10]:
```
no = [11, 30, 40, 43, 33 ,14 ,47, 57, 10 ,34 ,47 ,56, 36, 32 ,37 ,14 ,45, 35, 1
 26, 54, 10, 40, 43, 16, 18, 54, 56, 25, 28, 13, 18, 17, 19, 42, 45, 22 ,14, 51
 21, 21]
l=[10,20,30,40,50,60]
plt.hist(no,color='g',bins=l,edgecolor= "r")
plt.title('population')
plt.xlabel("python")
plt.ylabel("NO.")
plt.show()
```



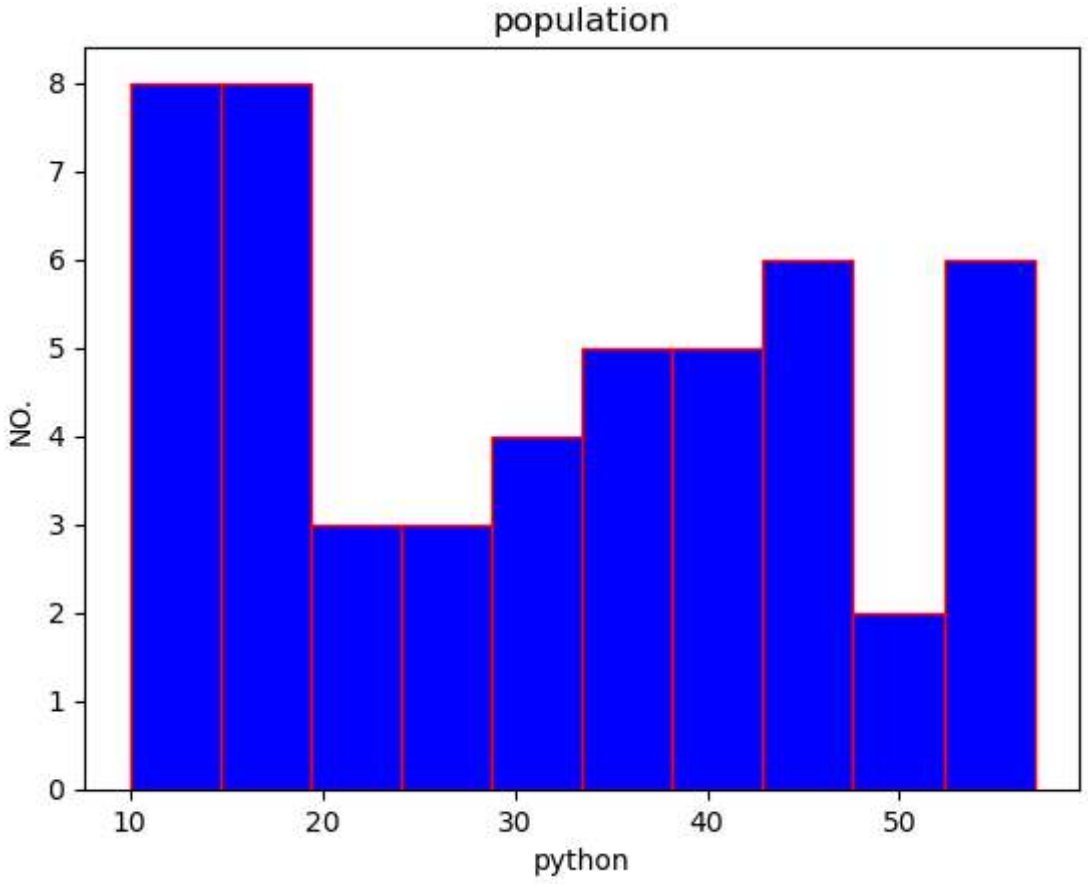
In [17]:
```
no = [11, 30, 40, 43, 33 ,14 ,47, 57, 10 ,34 ,47 ,56, 36, 32 ,37 ,14 ,45, 35, 1
 26, 54, 10, 40, 43, 16, 18, 54, 56, 25, 28, 13, 18, 17, 19, 42, 45, 22 ,14, 51
 21, 21]
l=[10,20,30,40,50,60]
plt.hist(no,color='b',edgecolor="r",cumulative=-1,bottom = 10)
plt.title('population')
plt.xlabel("python")
plt.ylabel("NO.")
plt.show()
```



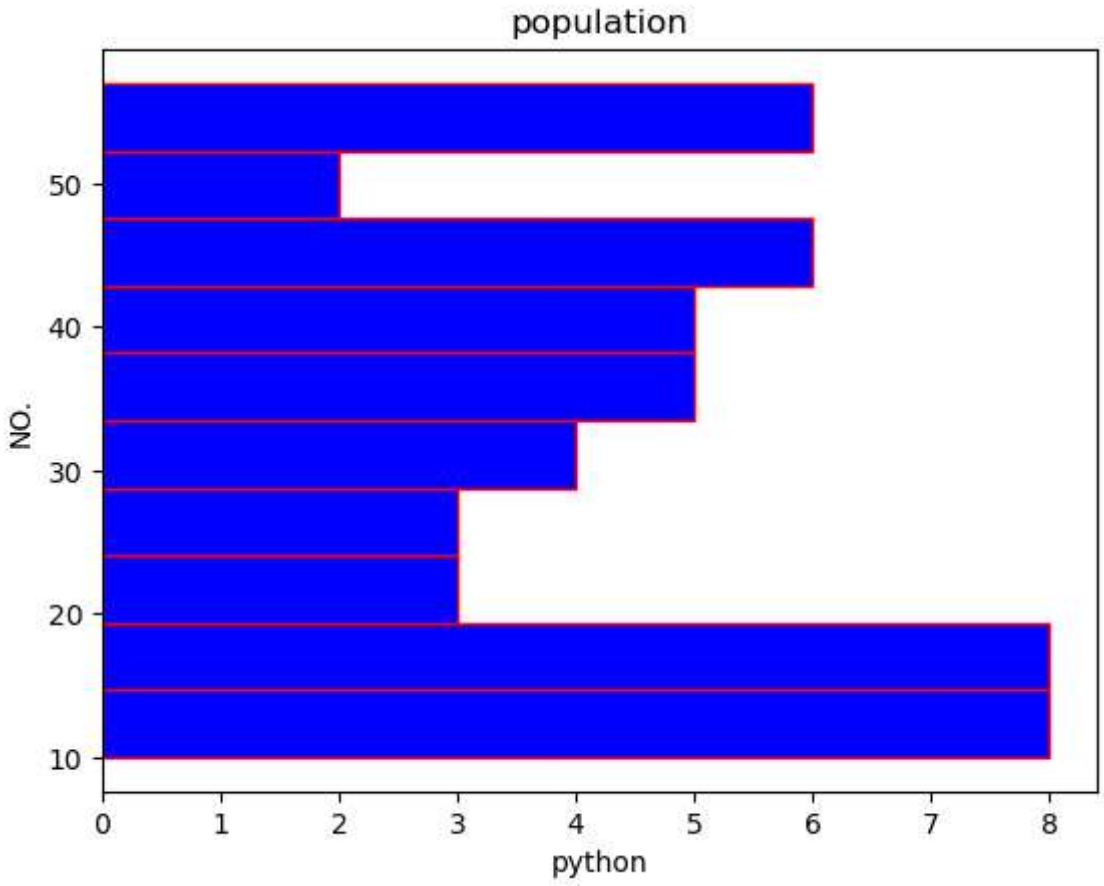
In [19]:
```
no = [11, 30, 40, 43, 33 ,14 ,47, 57, 10 ,34 ,47 ,56, 36, 32 ,37 ,14 ,45, 35, 1
 26, 54, 10, 40, 43, 16, 18, 54, 56, 25, 28, 13, 18, 17, 19, 42, 45, 22 ,14, 51
 21, 21]
l=[10,20,30,40,50,60]
plt.hist(no,color='b',edgecolor="r",histtype='step')
plt.title('population')
plt.xlabel("python")
plt.ylabel("NO.")
plt.show()
```



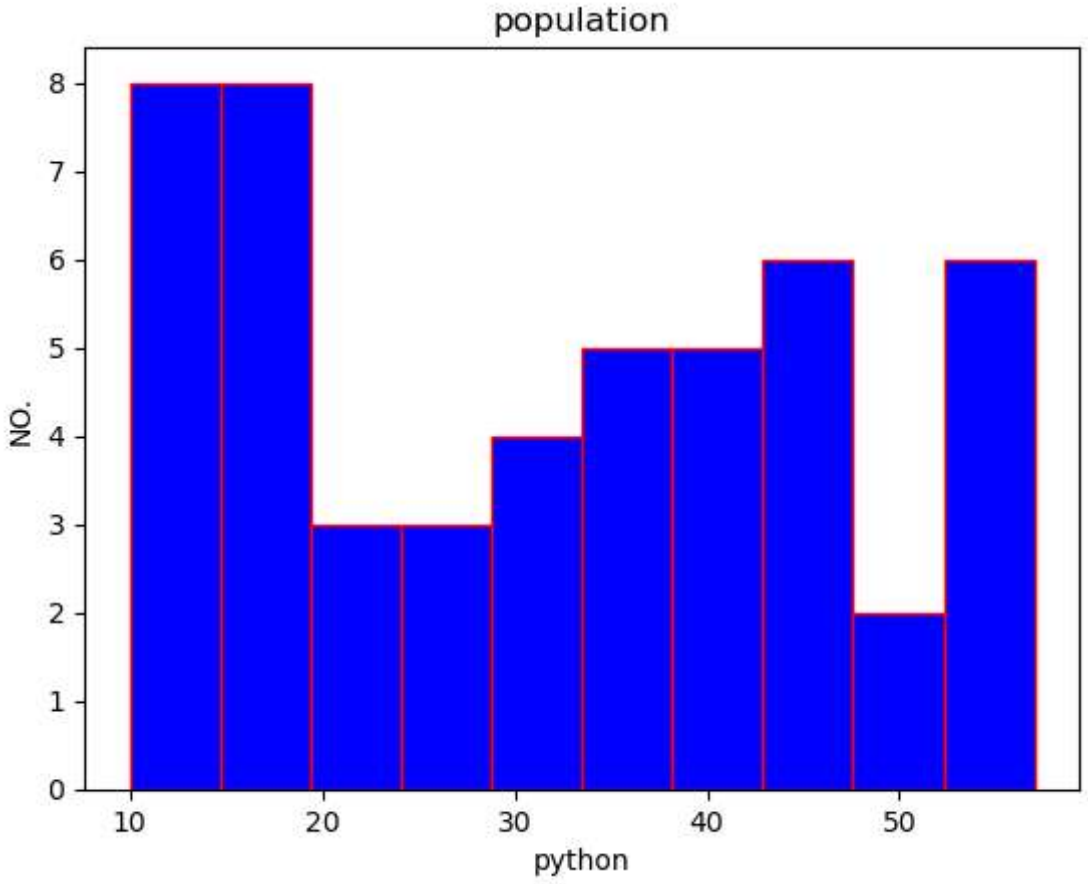
In [20]:
```
no = [11, 30, 40, 43, 33 ,14 ,47, 57, 10 ,34 ,47 ,56, 36, 32 ,37 ,14 ,45, 35, 1
 26, 54, 10, 40, 43, 16, 18, 54, 56, 25, 28, 13, 18, 17, 19, 42, 45, 22 ,14, 51
 21, 21]
l=[10,20,30,40,50,60]
plt.hist(no,color='b',edgecolor="r",histtype='barstacked')
plt.title('population')
plt.xlabel("python")
plt.ylabel("NO.")
plt.show()
```



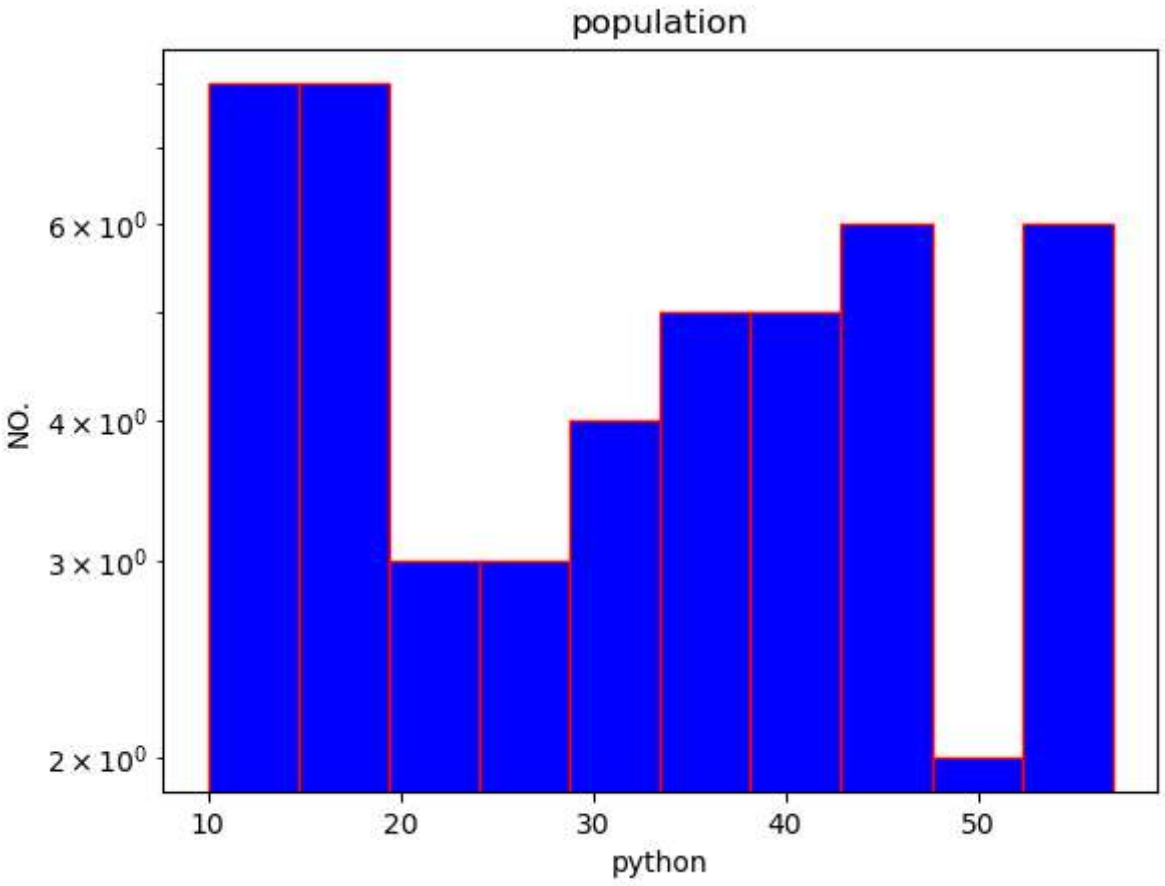
In [21]:
```
no = [11, 30, 40, 43, 33 ,14 ,47, 57, 10 ,34 ,47 ,56, 36, 32 ,37 ,14 ,45, 35, 1
 26, 54, 10, 40, 43, 16, 18, 54, 56, 25, 28, 13, 18, 17, 19, 42, 45, 22 ,14, 51
 21, 21]
l=[10,20,30,40,50,60]
plt.hist(no,color='b',edgecolor="r",orientation="horizontal")
plt.title('population')
plt.xlabel("python")
plt.ylabel("NO.")
plt.show()
```



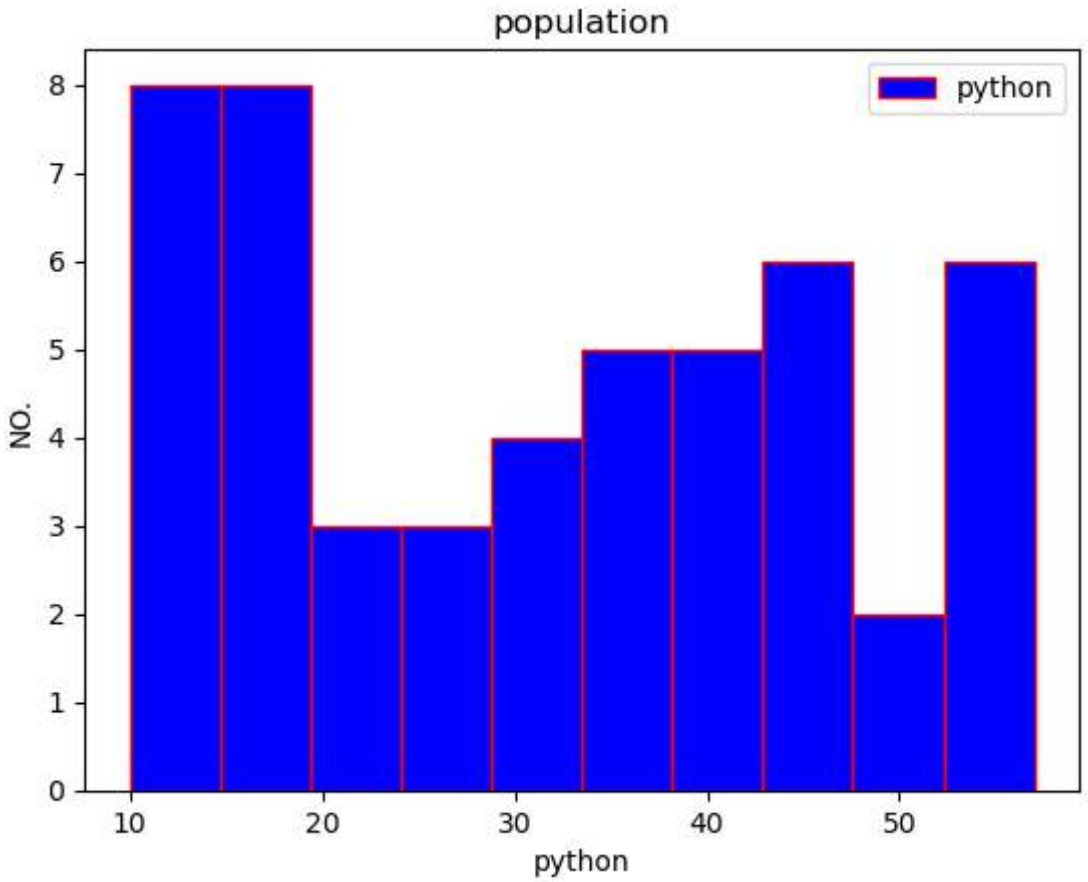
In [27]:
```
no = [11, 30, 40, 43, 33 ,14 ,47, 57, 10 ,34 ,47 ,56, 36, 32 ,37 ,14 ,45, 35, 1
 26, 54, 10, 40, 43, 16, 18, 54, 56, 25, 28, 13, 18, 17, 19, 42, 45, 22 ,14, 51
 21, 21]
l=[10,20,30,40,50,60]
plt.hist(no,color='b',edgecolor="r",orientation="vertical")
plt.title('population')
plt.xlabel("python")
plt.ylabel("NO.")
plt.show()
```



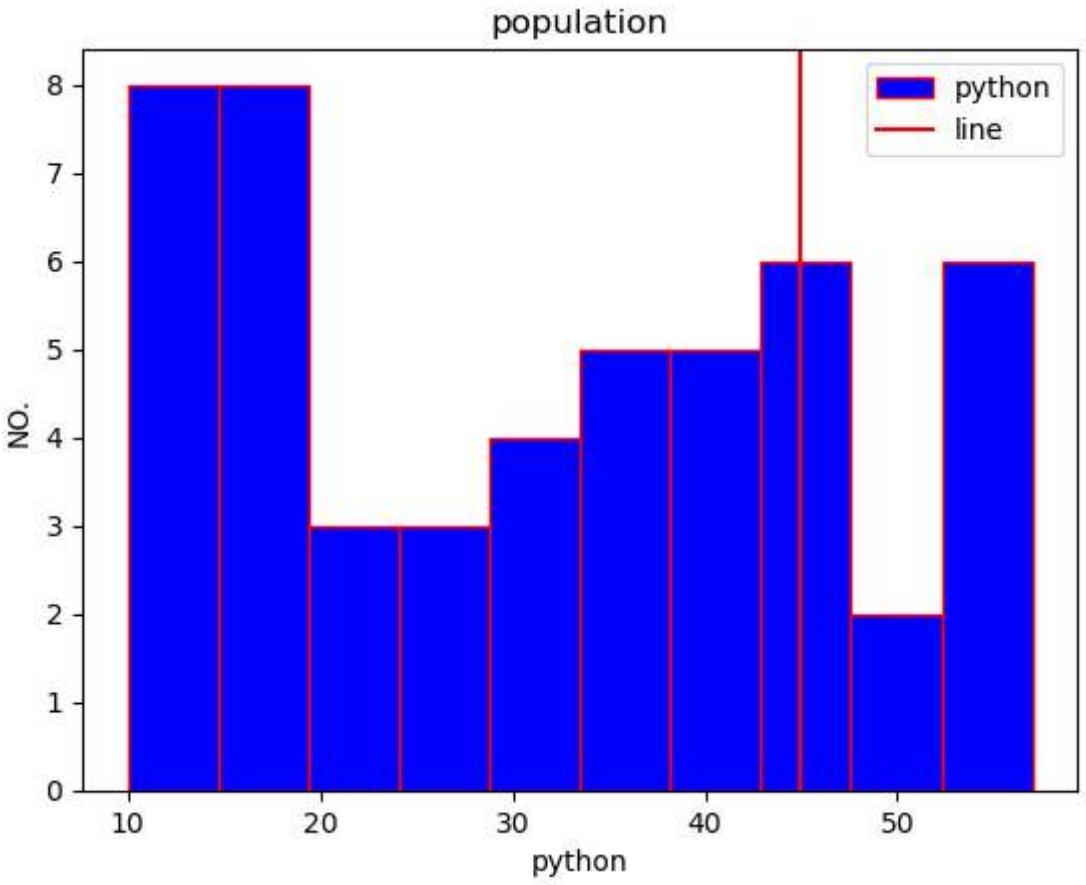
In [28]:
```
no = [11, 30, 40, 43, 33 ,14 ,47, 57, 10 ,34 ,47 ,56, 36, 32 ,37 ,14 ,45, 35, 1
 26, 54, 10, 40, 43, 16, 18, 54, 56, 25, 28, 13, 18, 17, 19, 42, 45, 22 ,14, 51
 21, 21]
l=[10,20,30,40,50,60]
plt.hist(no,color='b',edgecolor="r",log= True)
plt.title('population')
plt.xlabel("python")
plt.ylabel("NO.")
plt.show()
```



In [29]:
```
no = [11, 30, 40, 43, 33 ,14 ,47, 57, 10 ,34 ,47 ,56, 36, 32 ,37 ,14 ,45, 35, 1
 26, 54, 10, 40, 43, 16, 18, 54, 56, 25, 28, 13, 18, 17, 19, 42, 45, 22 ,14, 51
 21, 21]
l=[10,20,30,40,50,60]
plt.hist(no,color='b',edgecolor="r",label='python')
plt.title('population')
plt.xlabel("python")
plt.ylabel("NO.")
plt.legend()
plt.show()
```

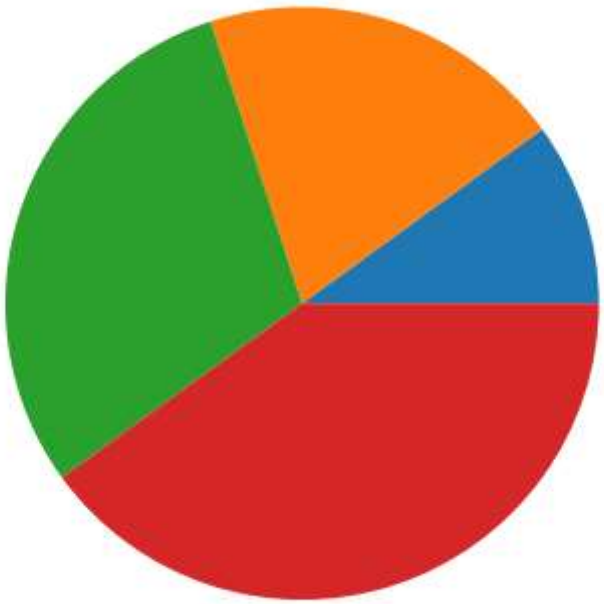


In [3]:
```
no = [11, 30, 40, 43, 33 ,14 ,47, 57, 10 ,34 ,47 ,56, 36, 32 ,37 ,14 ,45, 35, 1
 26, 54, 10, 40, 43, 16, 18, 54, 56, 25, 28, 13, 18, 17, 19, 42, 45, 22 ,14, 51
 21, 21]
l=[10,20,30,40,50,60]
plt.hist(no,color='b',edgecolor="r",label='python')
plt.title('population')
plt.xlabel("python")
plt.ylabel("NO.")
plt.axvline(45,color='r',label='line')
plt.legend()
plt.show()
```

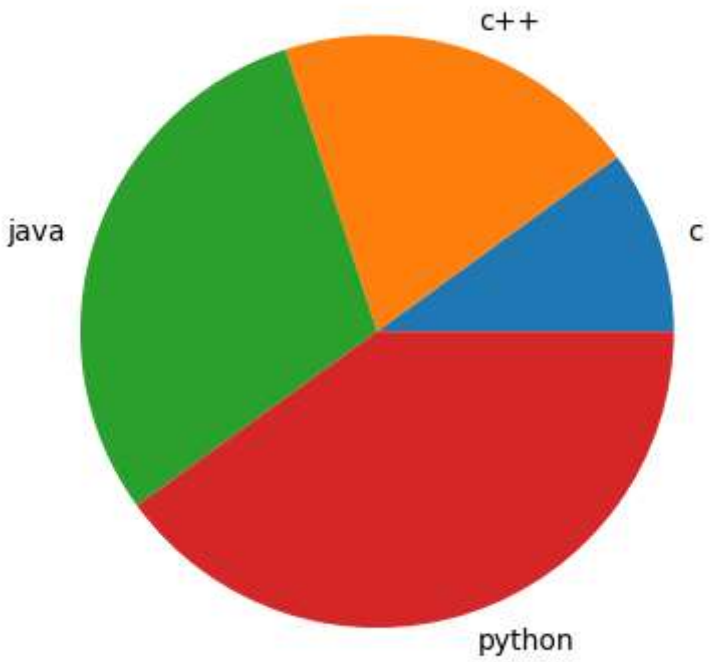


pie chart

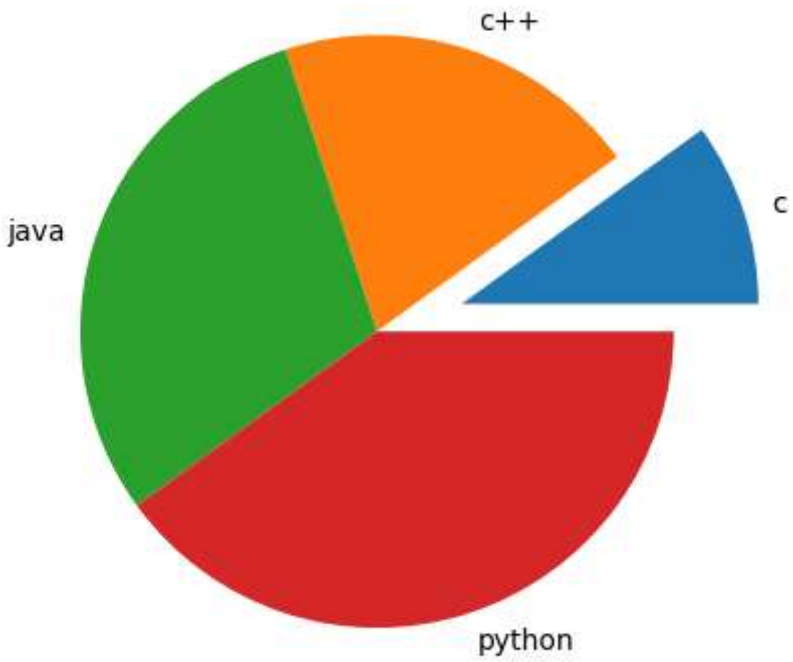
In [4]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']

plt.pie(x)

plt.show()
```



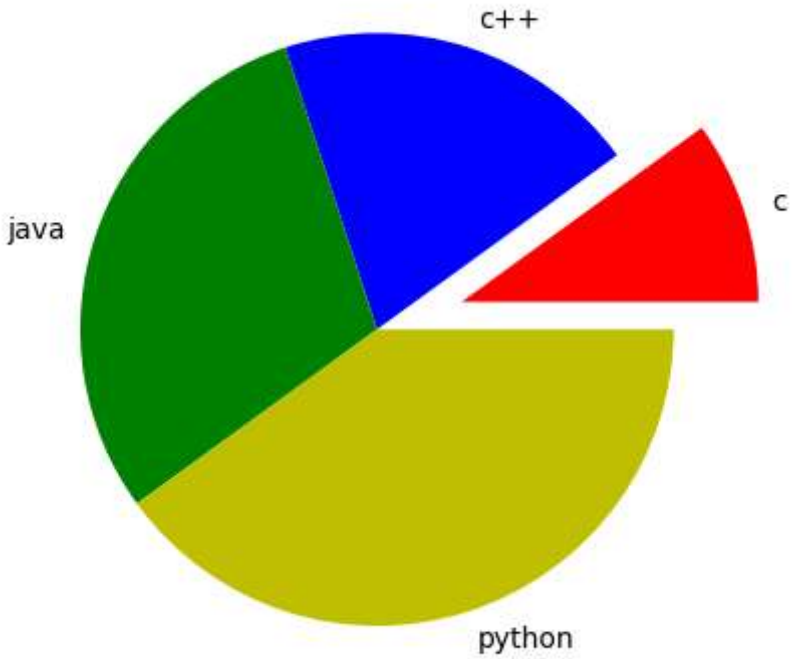
In [6]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']

plt.pie(x,labels=y)

plt.show()
```



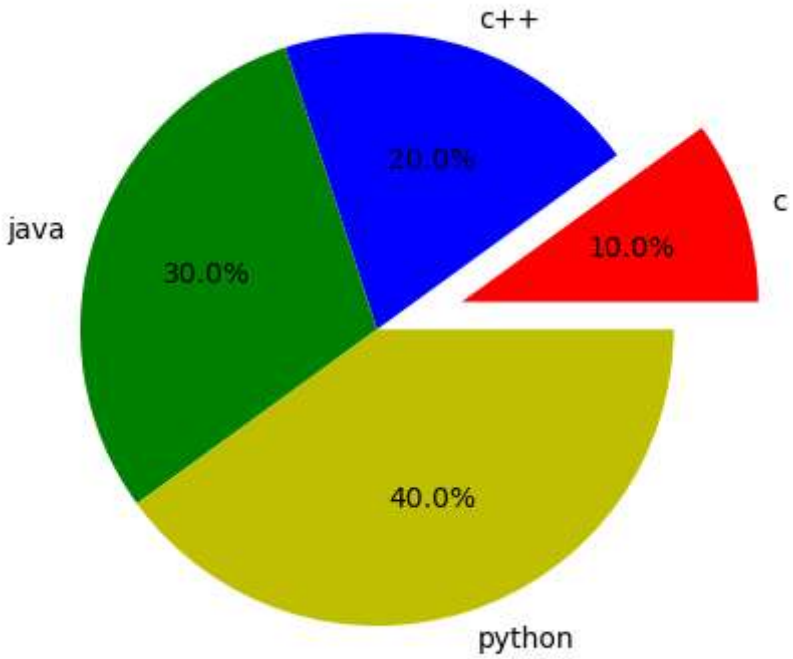
In [11]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]

plt.pie(x,labels=y,explode=ex)

plt.show()
```



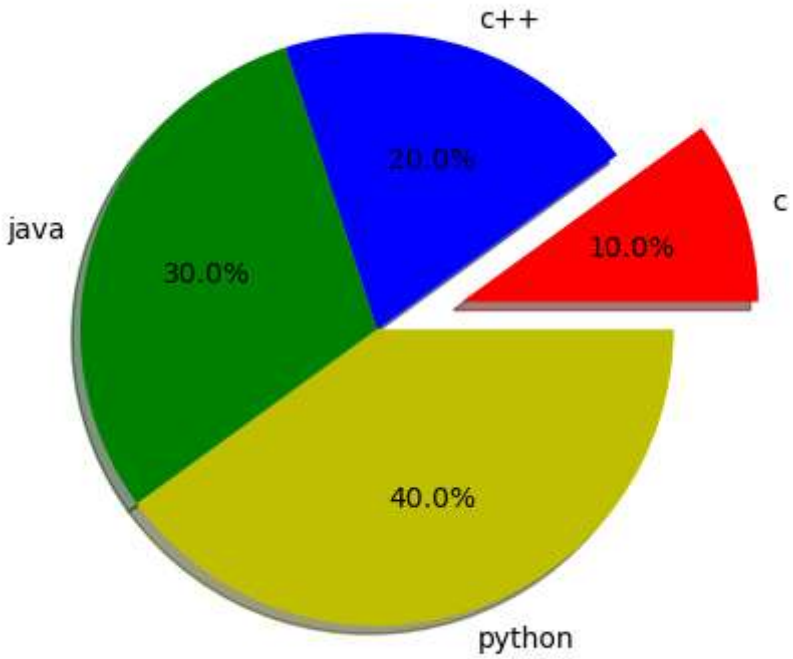
In [14]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c)

plt.show()
```



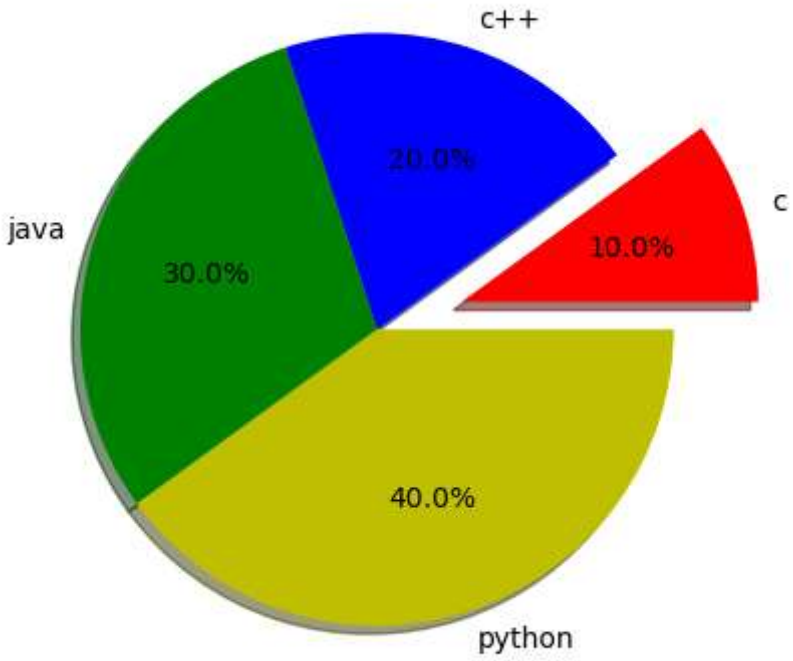
In [19]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%")

plt.show()
```



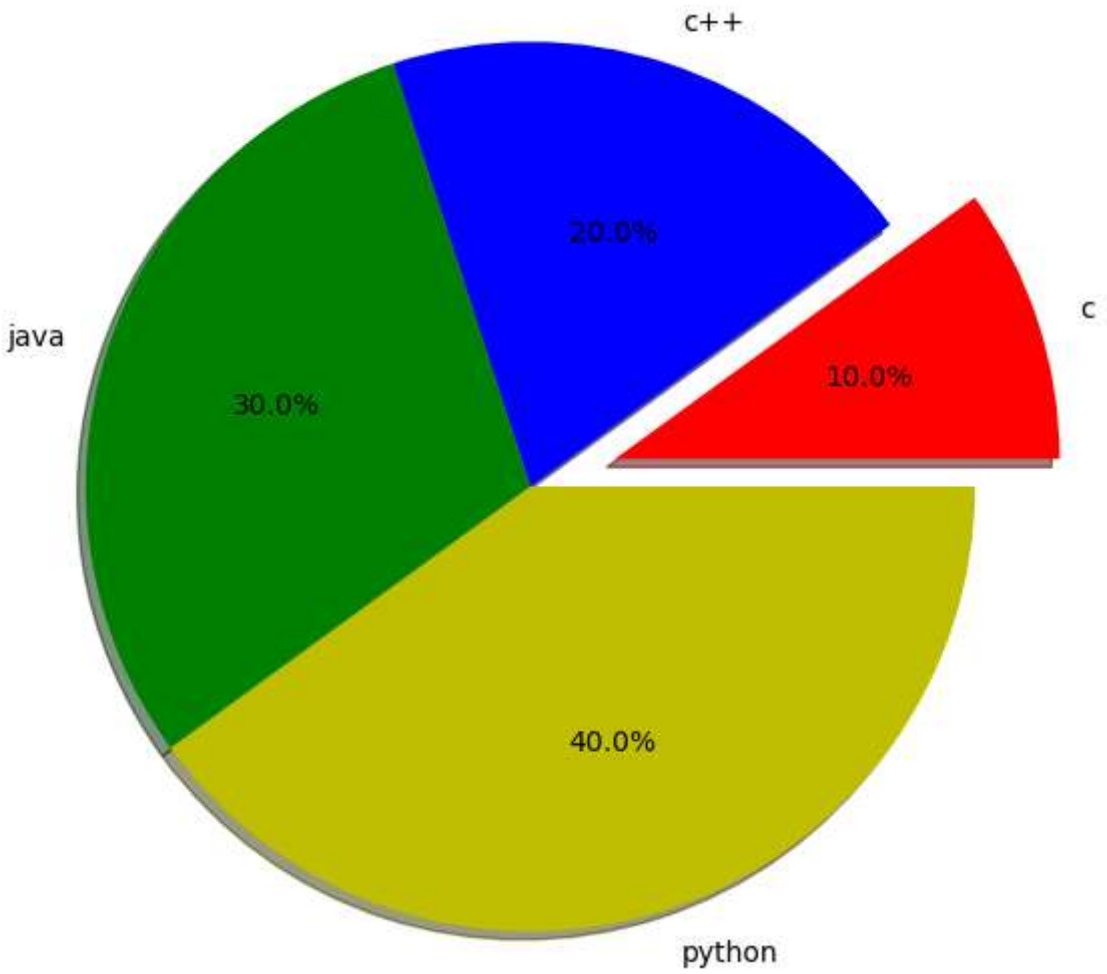
In [20]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",shadow=True)

plt.show()
```



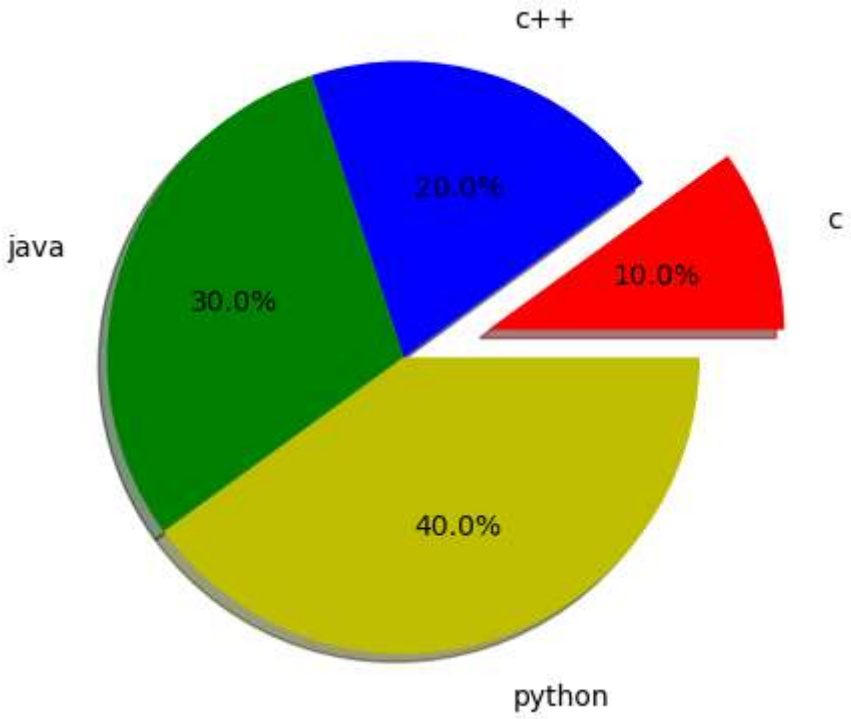
In [21]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",shadow=True,radius=1)

plt.show()
```



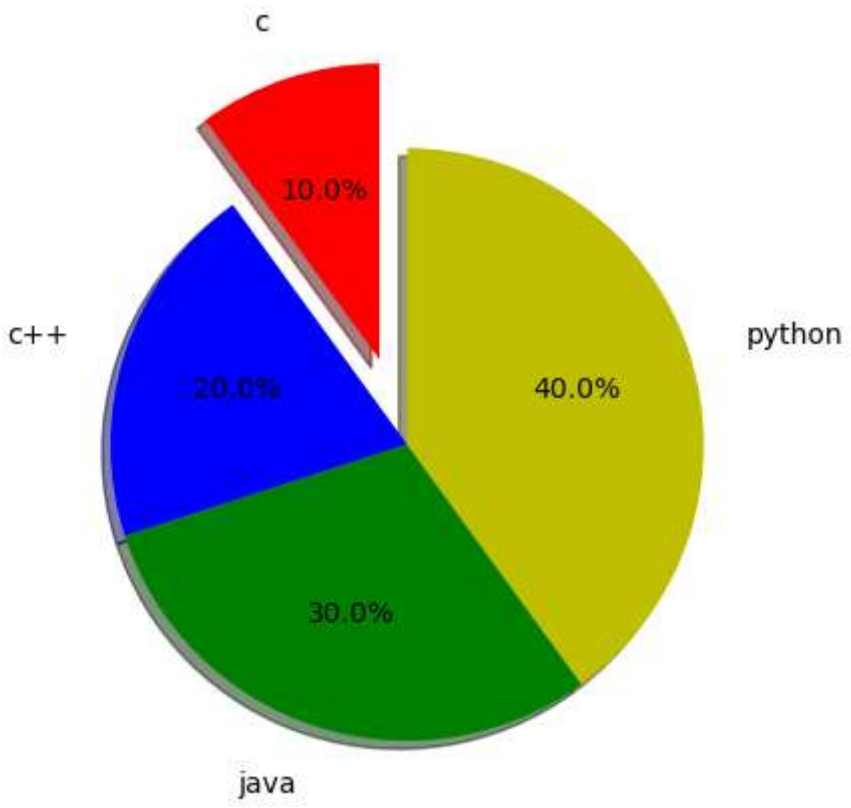
In [22]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",shadow=True,radius=1.5

plt.show()
```



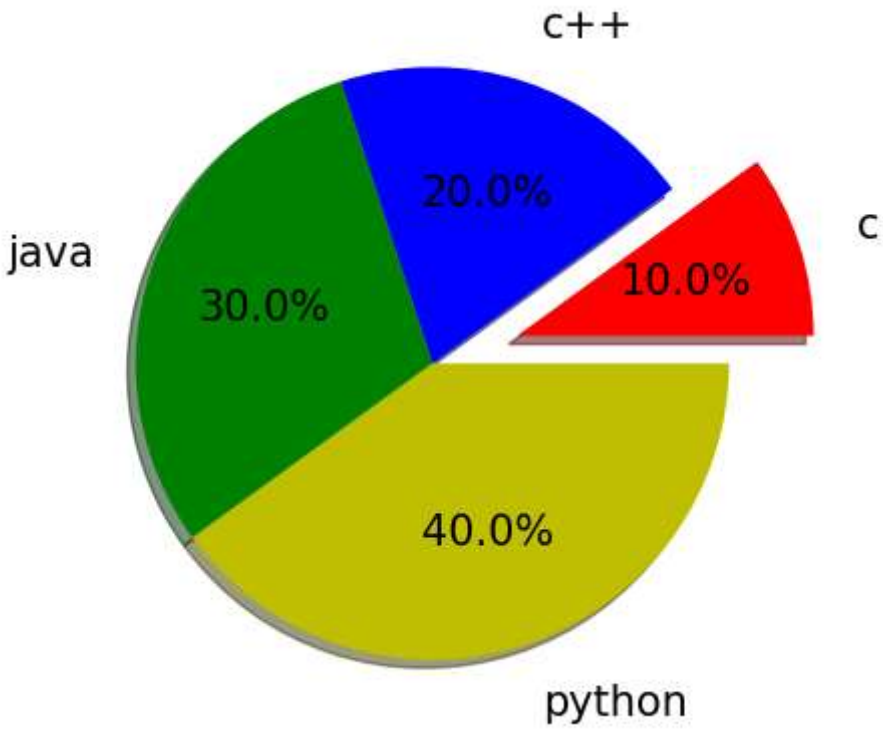
In [24]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",shadow=True,radius=1,

plt.show()
```



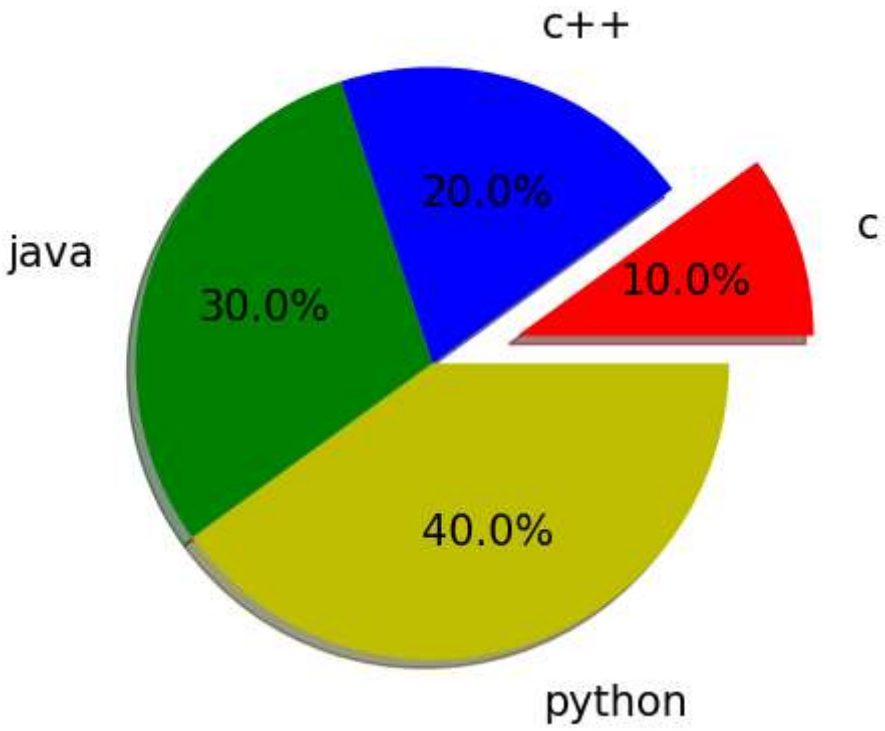
In [27]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",shadow=True,radius=1,

plt.show()
```



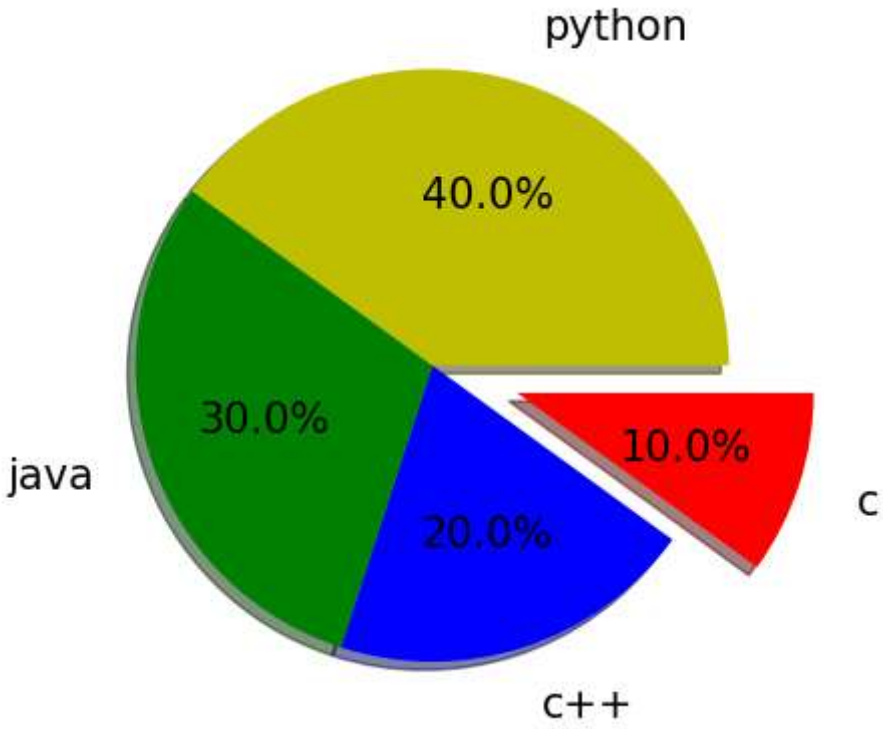
In [31]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",shadow=True,radius=1,

plt.show()
```



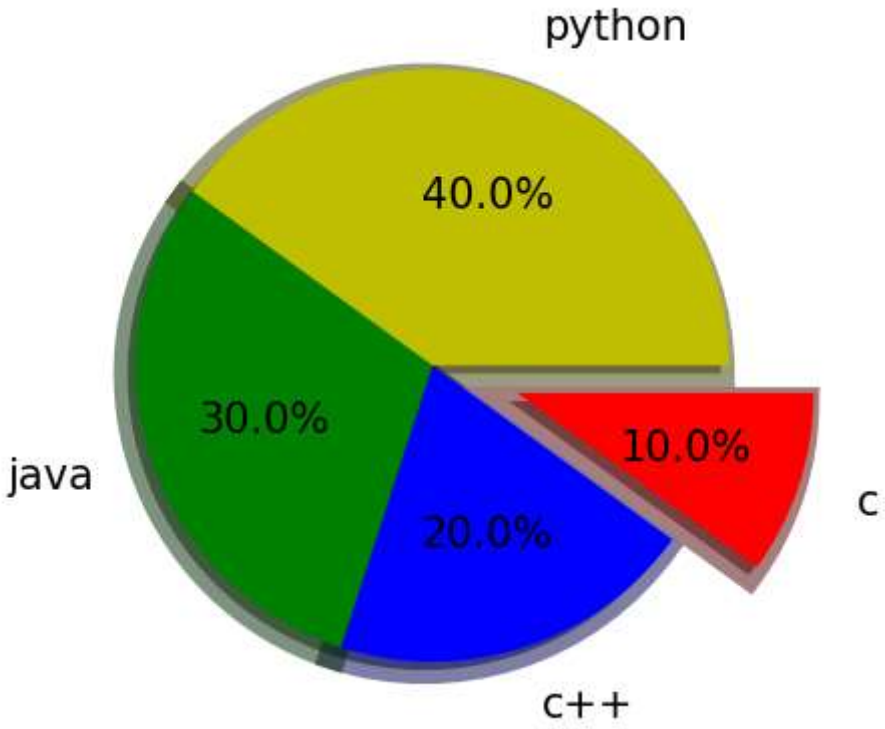
In [32]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",
        shadow=True,radius=1,labeldistance=1.2,textprops={'fontsize':15},counte

plt.show()
```



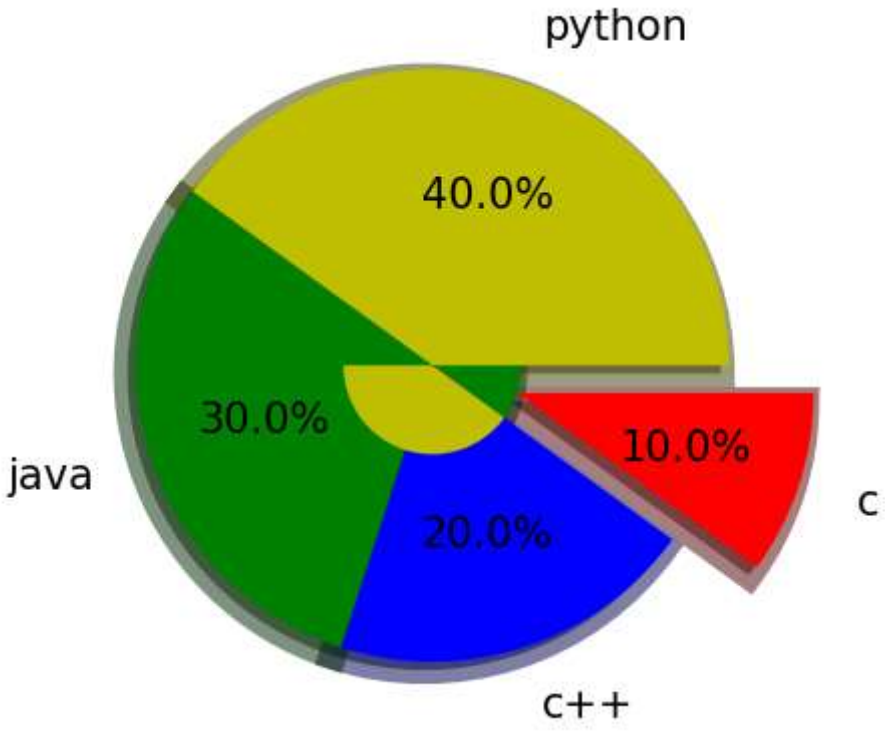
In [33]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",
        shadow=True,radius=1,labeldistance=1.2,textprops={'fontsize':15},counte

plt.show()
```



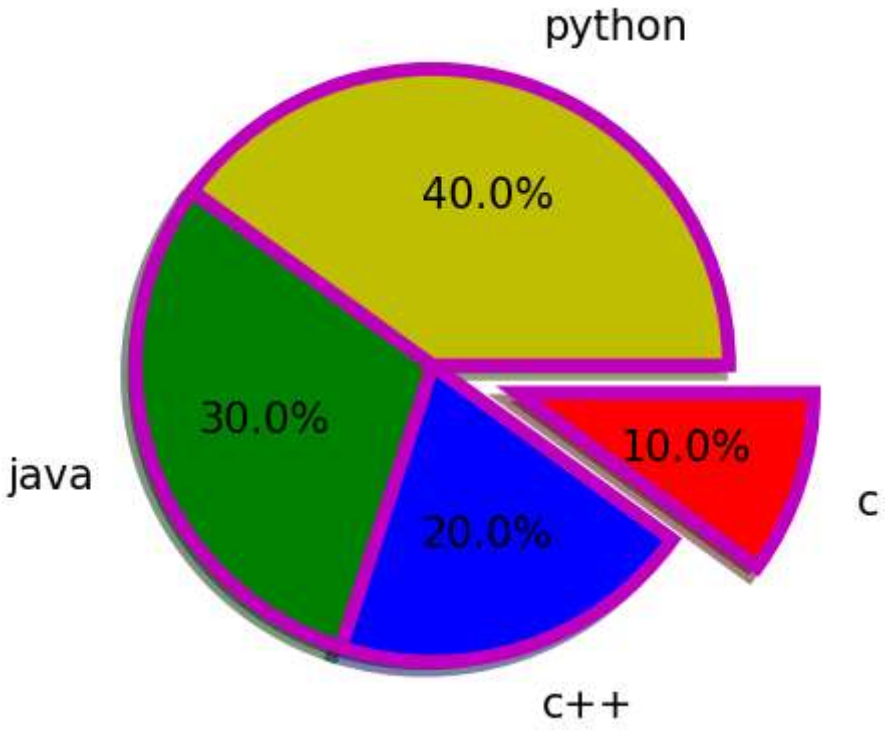
In [36]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",
        shadow=True,radius=1,labeldistance=1.2,textprops={'fontsize':15},counte

plt.show()
```



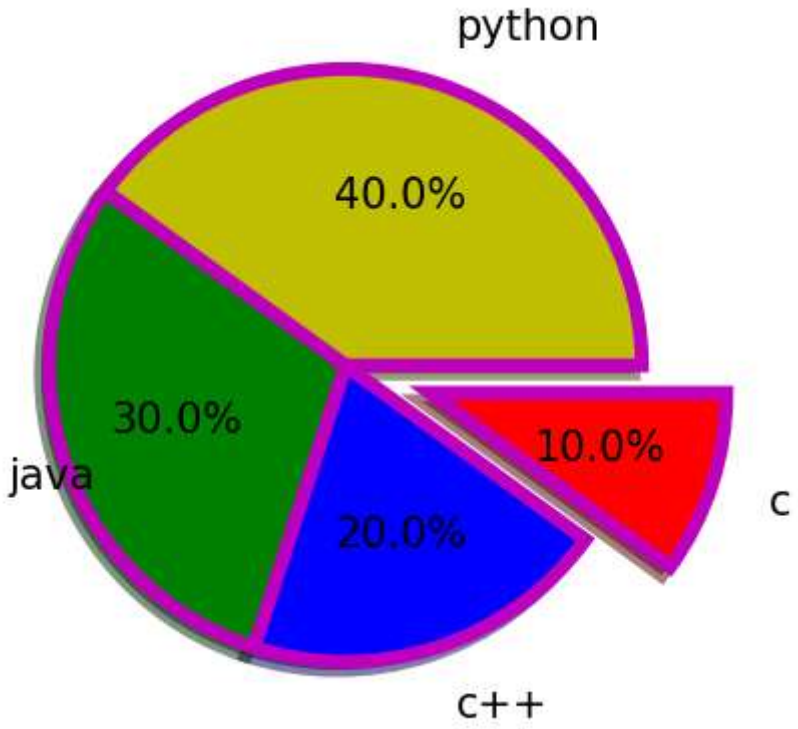
In [38]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",
        shadow=True,radius=1,labeldistance=1.2,textprops={'fontsize':15},
        counterclock=False,wedgeprops={'linewidth':10,'width':1.3})

plt.show()
```



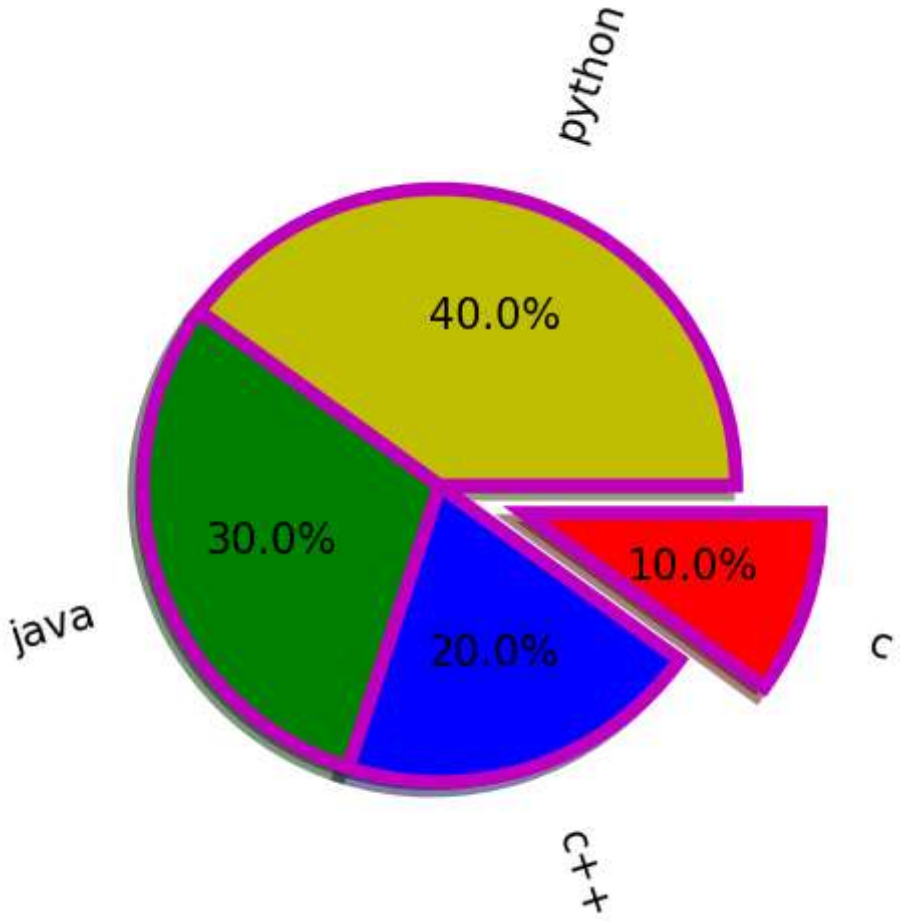
In [40]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",
        shadow=True,radius=1,labeldistance=1.2,textprops={'fontsize':15},
        counterclock=False,wedgeprops={'linewidth':5,'edgecolor':'m'})

plt.show()
```



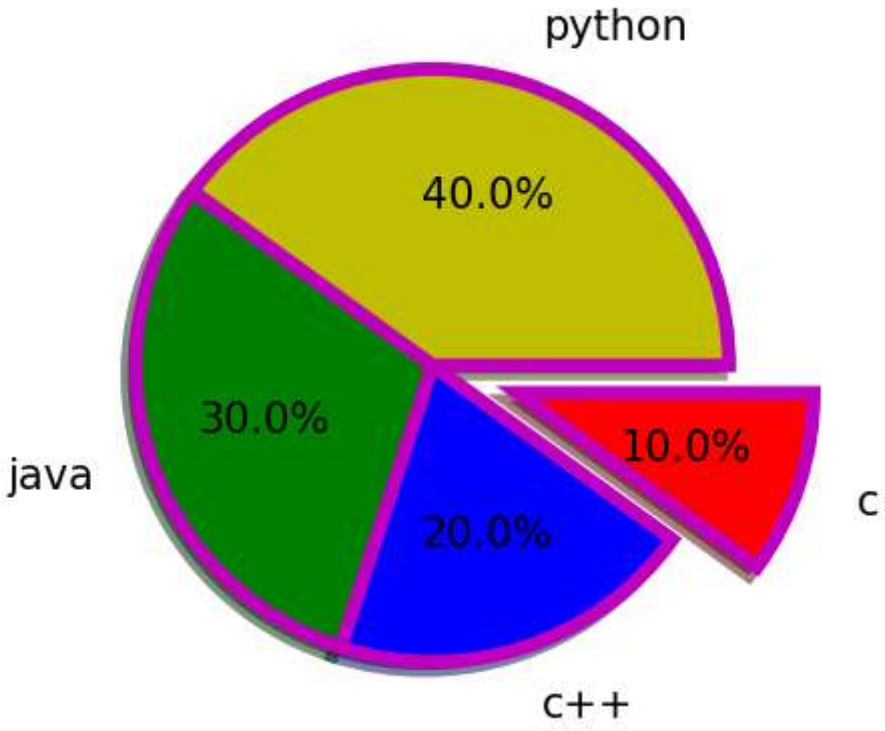
In [45]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",
        shadow=True,radius=1,labeldistance=1.2,textprops={'fontsize':15},
        counterclock=False,wedgeprops={'linewidth':5,'edgecolor':'m'},center=(

plt.show()
```



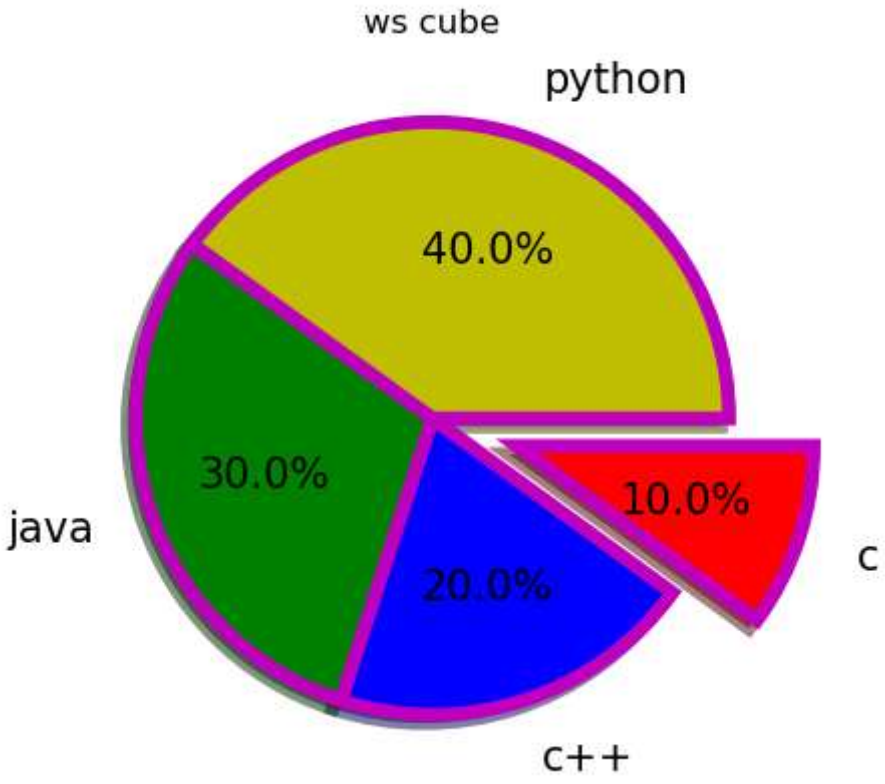
In [47]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",
        shadow=True,radius=1,labeldistance=1.2,textprops={'fontsize':15},
        counterclock=False,wedgeprops={'linewidth':5,'edgecolor':'m'},rotatelab

plt.show()
```



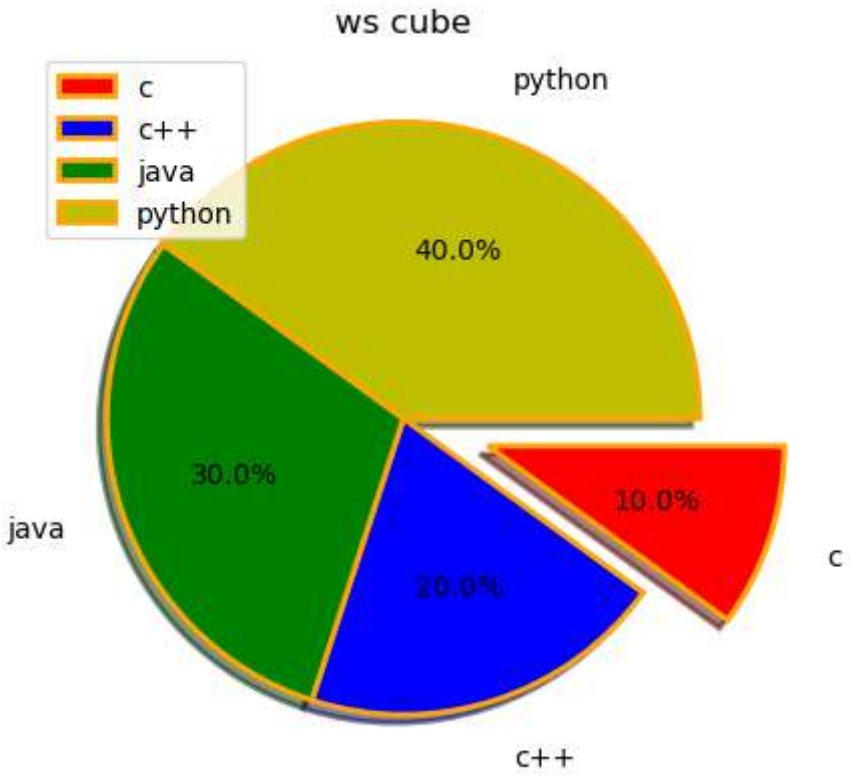
In [48]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",
        shadow=True,radius=1,labeldistance=1.2,textprops={'fontsize':15},
        counterclock=False,wedgeprops={'linewidth':5,'edgecolor':'m'},rotatelab

plt.show()
```



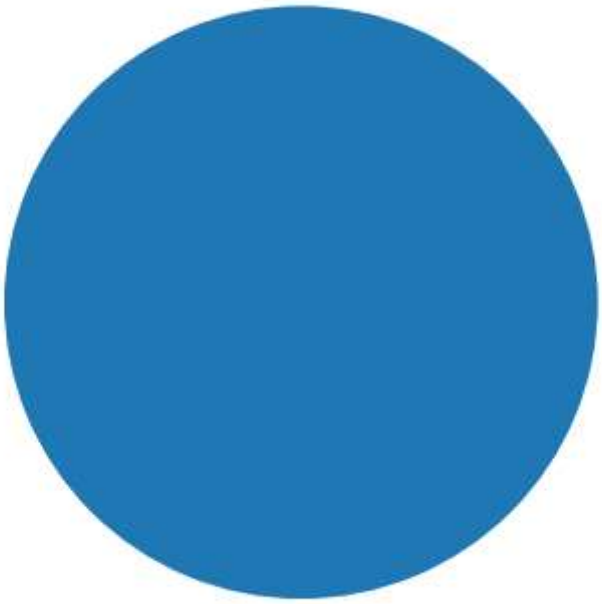
In [49]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",
        shadow=True,radius=1,labeldistance=1.2,textprops={'fontsize':15},
        counterclock=False,wedgeprops={'linewidth':5,'edgecolor':'m'},rotatelab
plt.title("ws cube")
plt.show()
```



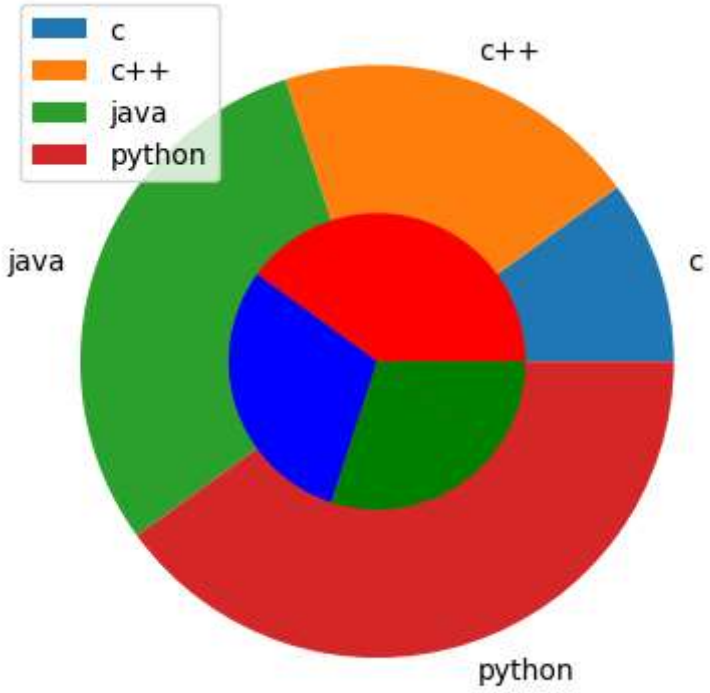
In [53]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
ex = [0.3,0.0,0.0,0.0]
c=["r","b","g","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",
        shadow=True,radius=1,labeldistance=1.2,textprops={'fontsize':10},
        counterclock=False,wedgeprops={'linewidth':2,'edgecolor':'orange'},rota
plt.title("ws cube")
plt.legend(loc=2)
plt.show()
```



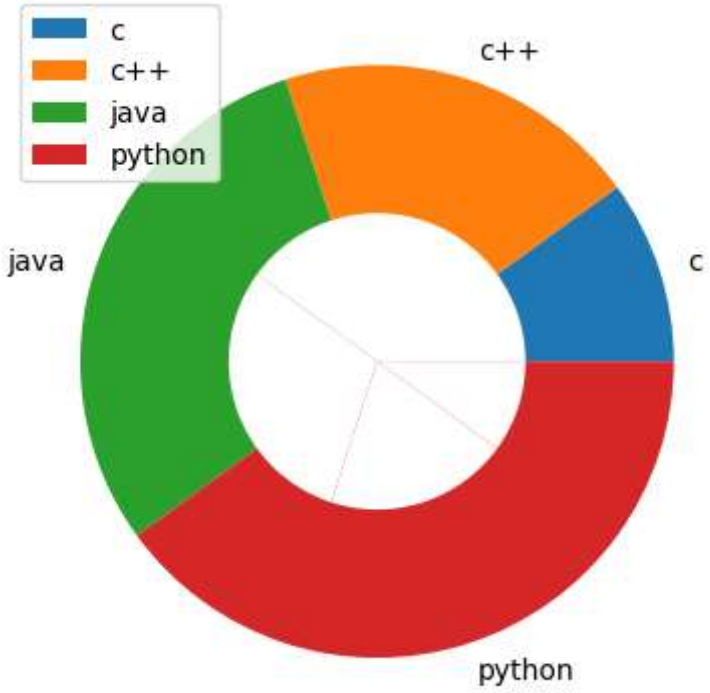
In [54]:
```
plt.pie([1])
plt.show()
```



In [61]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
x1 = [40,30,20,10]
c=["r","b","g","g"]
plt.pie(x,labels=y,radius=1.0)
plt.pie(x1,radius=0.5,colors=c)

plt.legend(loc=2)
plt.show()
```

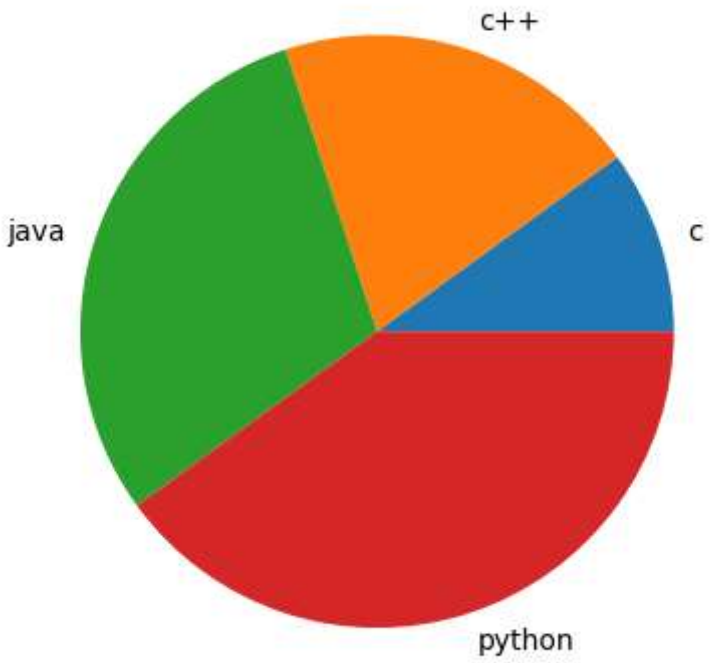


In [62]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
x1 = [40,30,20,10]
c=["r","b","g","g"]
plt.pie(x,labels=y,radius=1.0)
plt.pie(x1,radius=0.5,colors='w')

plt.legend(loc=2)
plt.show()
```



In [63]:
```
x = [10,20,30,40]
y = ['c','c++','java','python']
x1 = [40,30,20,10]
c=["r","b","g","g"]
plt.pie(x,labels=y,radius=1.0)
cr=plt.circle(xy=(0,0),radius=1,facecolor="w")
plt.gca().add_artist(cr)

plt.legend(loc=2)
plt.show()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[63], line 6
      4 c=["r","b","g","g"]
      5 plt.pie(x,labels=y,radius=1.0)
----> 6 cr=plt.circle(xy=(0,0),radius=1,facecolor="w")
      7 plt.gca().add_artist(cr)
      9 plt.legend(loc=2)

AttributeError: module 'matplotlib.pyplot' has no attribute 'circle'
```



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

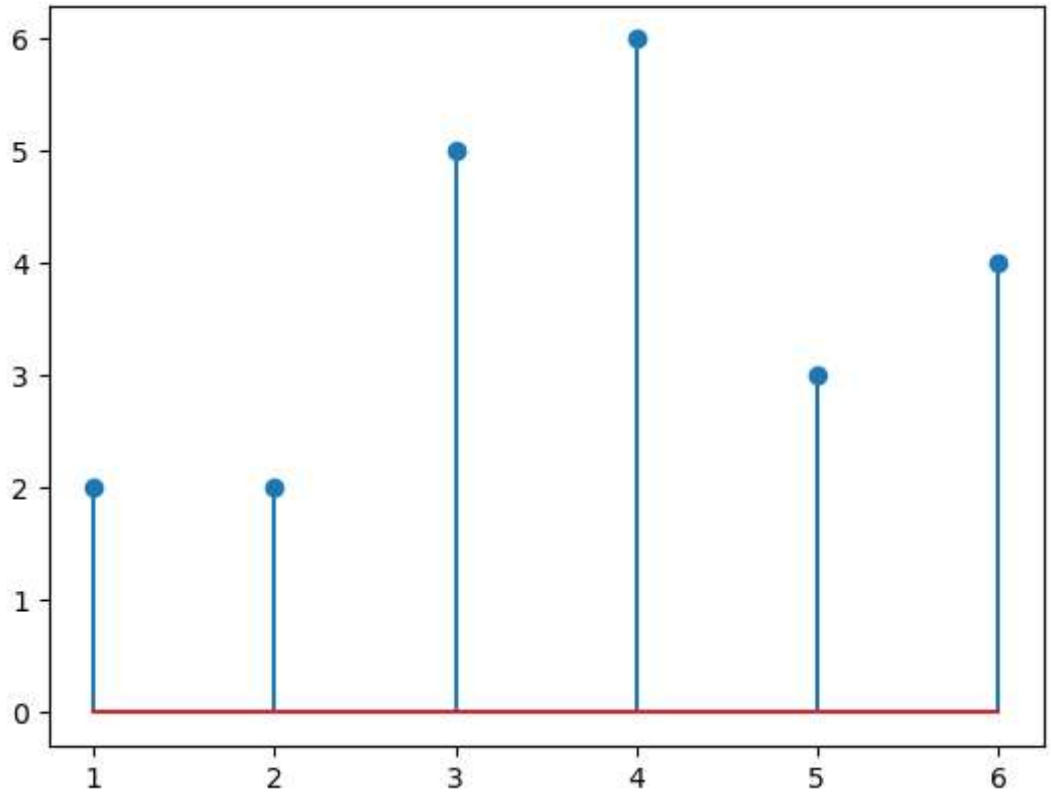
stem plot

x = []

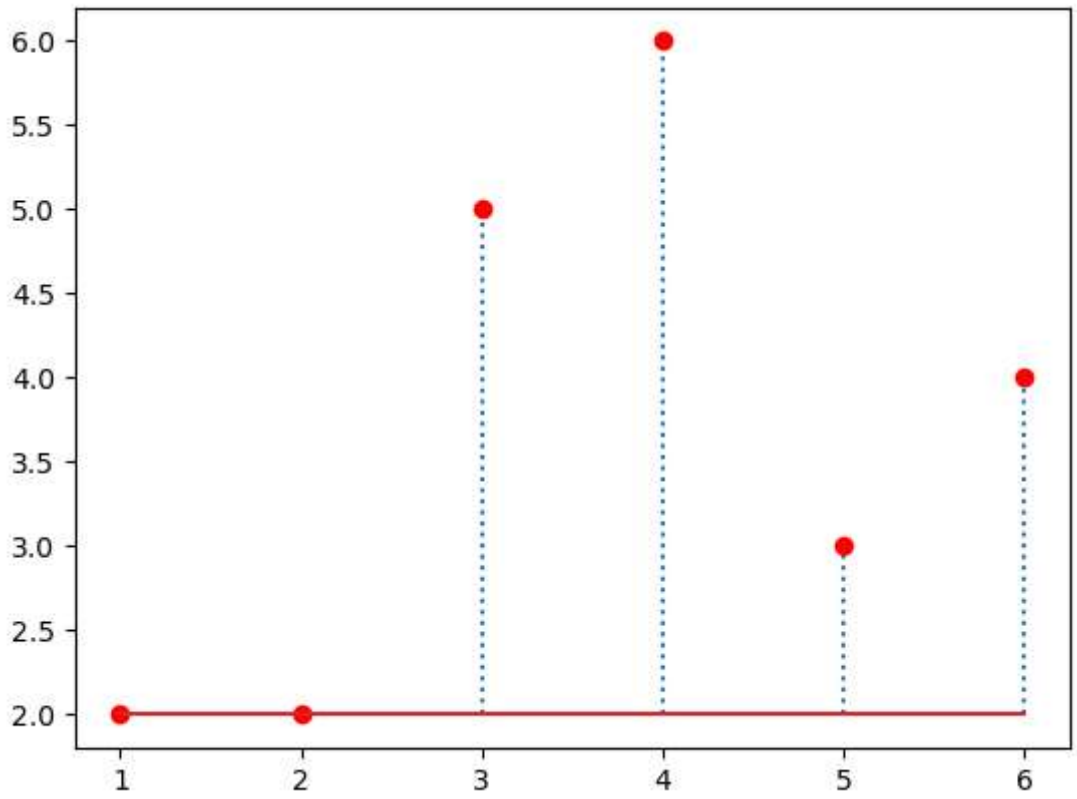
y = []

plt.stem(x,y)

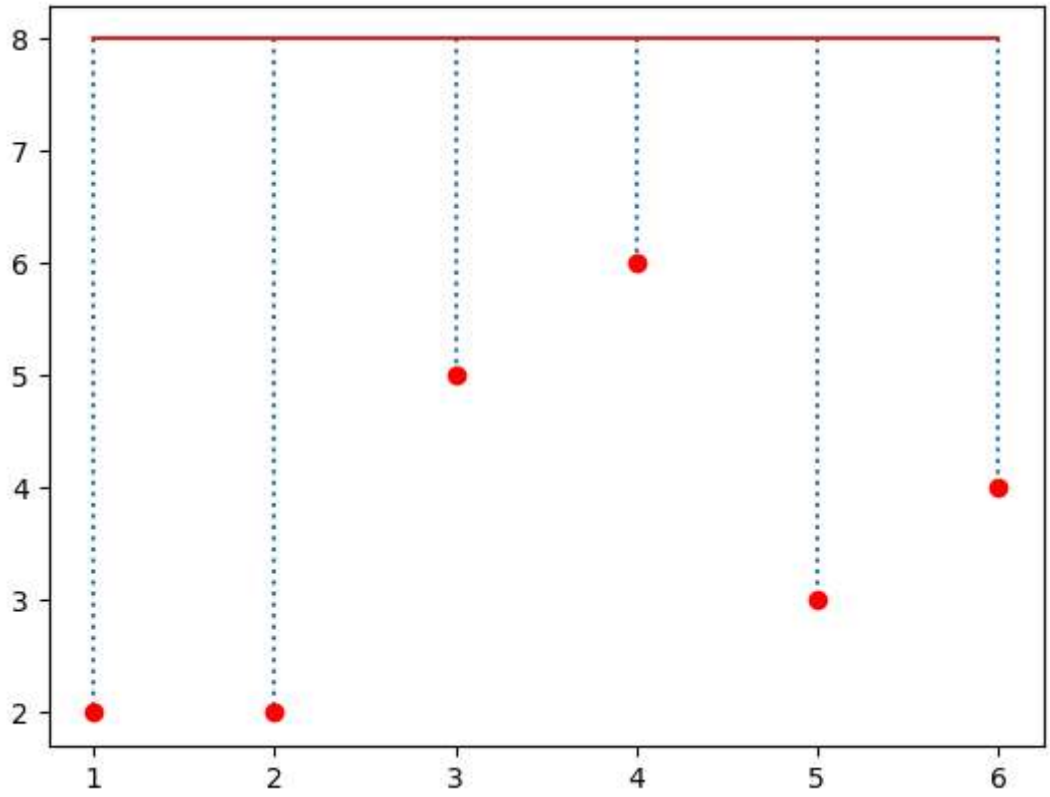
In [7]:
```
x = [1,2,3,4,5,6]
y=[2,2,5,6,3,4]
plt.stem(x,y)
plt.show()
```



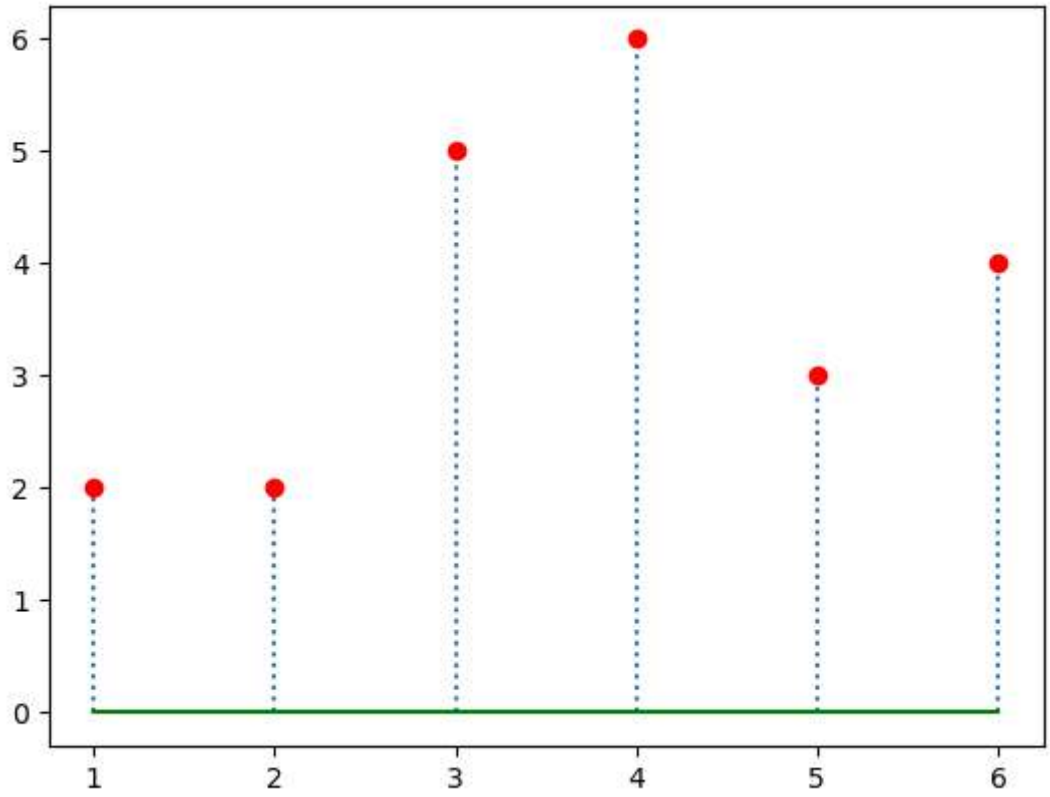
In [15]:
```
x = [1,2,3,4,5,6]
y=[2,2,5,6,3,4]
plt.stem(x,y,linefmt=':',markerfmt='ro',bottom=2)
plt.show()
```



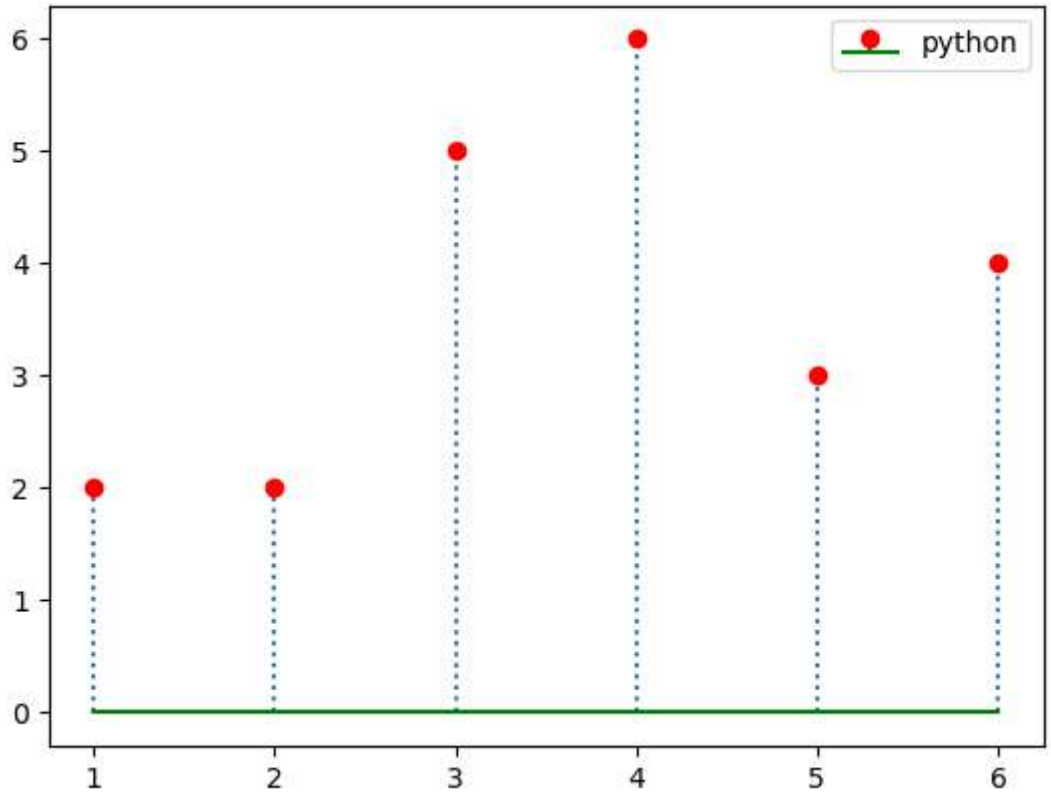
In [16]:
```
x = [1,2,3,4,5,6]
y=[2,2,5,6,3,4]
plt.stem(x,y,linefmt=':',markerfmt='ro',bottom=8)
plt.show()
```



In [17]:
```
x = [1,2,3,4,5,6]
y=[2,2,5,6,3,4]
plt.stem(x,y,linefmt=':',markerfmt='ro',basefmt='g')
plt.show()
```

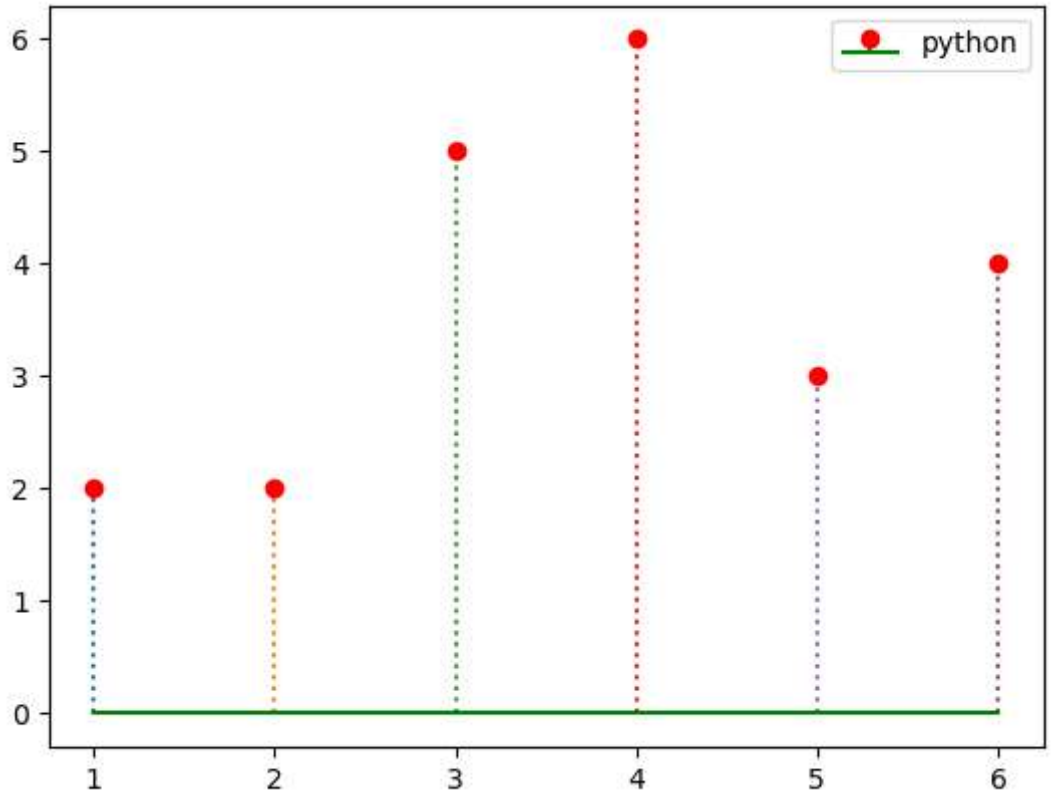


In [18]:
```
x = [1,2,3,4,5,6]
y=[2,2,5,6,3,4]
plt.stem(x,y,linefmt=':',markerfmt='ro',basefmt='g',label='python')
plt.legend()
plt.show()
```



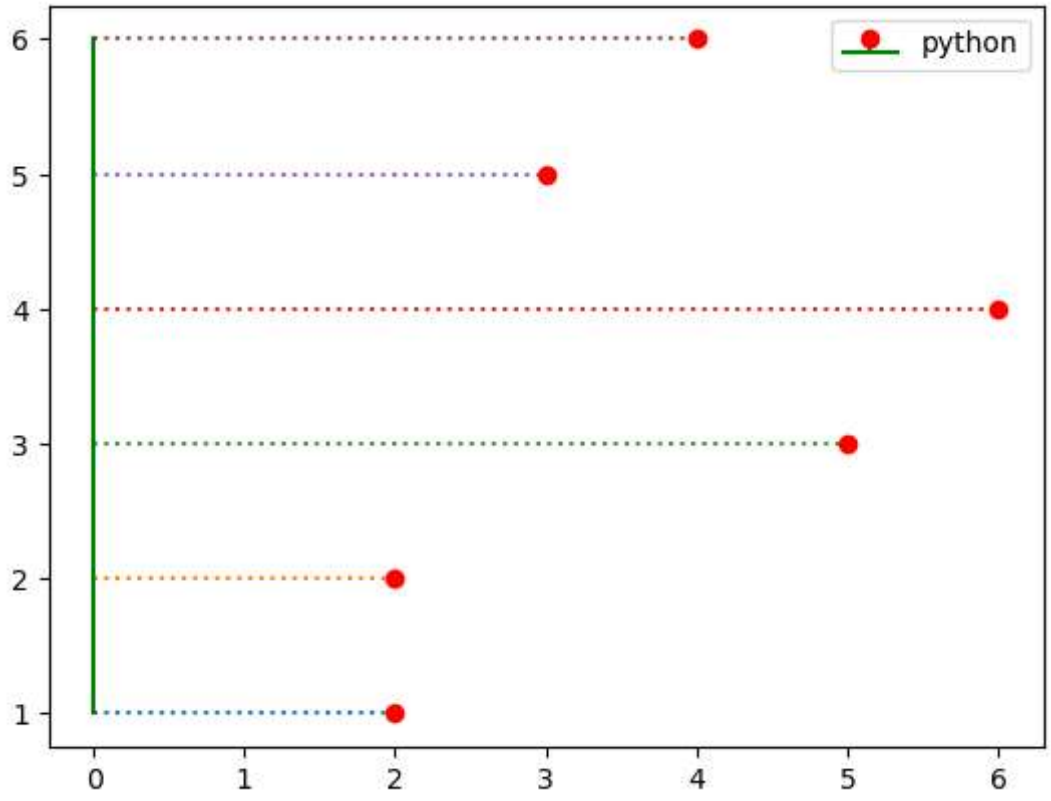
In [19]:
```
x = [1,2,3,4,5,6]
y=[2,2,5,6,3,4]
plt.stem(x,y,linefmt=':',markerfmt='ro',basefmt='g',label='python',use_line_co
plt.legend()
plt.show()
```

```
C:\Users\Akhilesh\AppData\Local\Temp\ipykernel_1104\1445273911.py:3: Matplotl
ibDeprecationWarning: The 'use_line_collection' parameter of stem() was depre
cated in Matplotlib 3.6 and will be removed two minor releases later. If any
parameter follows 'use_line_collection', they should be passed as keyword, no
t positionally.
  plt.stem(x,y,linefmt=':',markerfmt='ro',basefmt='g',label='python',use_line
_collection=False)
```



In [20]:
```
x = [1,2,3,4,5,6]
y=[2,2,5,6,3,4]
plt.stem(x,y,linefmt=':',markerfmt='ro',basefmt='g',label='python',use_line_co
plt.legend()
plt.show()
```

```
C:\Users\Akhilesh\AppData\Local\Temp\ipykernel_1104\473533506.py:3: MatplotlibDeprecationWarning: The 'use_line_collection' parameter of stem() was deprecated in Matplotlib 3.6 and will be removed two minor releases later. If any parameter follows 'use_line_collection', they should be passed as keyword, not positionally.
  plt.stem(x,y,linefmt=':',markerfmt='ro',basefmt='g',label='python',use_line_collection=False , orientation='horizontal')
```



In [ ]: