In [64]:

```python
import matplotlib.pyplot as plt
import scipy.io
from random import randint
import random
import numpy as np
import math
datafile = 'Kmeansdata.mat'
points = scipy.io.loadmat( datafile )
#x=np.array(points)
t=points['X']
x=t[:, 0]
y=t[:, 1]
```

In [65]:

```python
def rand_centroid(x,y):
    n1=random.uniform(np.amin(x),np.amax(x))
    n2=random.uniform(np.amin(y),np.amax(y))
    return n1,n2
```

In [66]:

```python
def distance(n,o):
    d=math.sqrt(((n[0]-o[0])**2)+((n[1]-o[1])**2))
    return d
```

In [67]:

```python
#this function is used to classifi the data set in centroid groups
def classify(t,m):
    newt=[]
    for i in range(len(t)):
        dis=[]
        for j in range(len(m)):
            dis.append(distance(m[j],t[i]))
        #    print(dis[j])
        newt.append(dis.index(np.amin(dis)))
    return newt
```

In [68]:

```python
def new_cent(t,newt):
    for k in range(3):
        rat=[]
        rat1=[]

        for i in range(len(newt)):
            if newt[i]==k:
        #       plt.scatter(v[i],y[i],'o', color='black')
                rat.append(x[i])
                rat1.append(y[i])
        m[k,0]=np.mean(rat)
        m[k,1]=np.mean(rat1)
        #plt.scatter(rat,rat1)
        #print(np.mean(rat))
        #return rat,rat1
    #plt.scatter(m[:,0],m[:,1])
    return m
```

In [69]:

```
def k_centroid():
    lo=[]
    for i in range(3):
        r=rand_centroid(x,y)
        #plt.scatter(r[0],r[1])
        lo.append(r)
    #plt.scatter(x,y)
    return lo
```

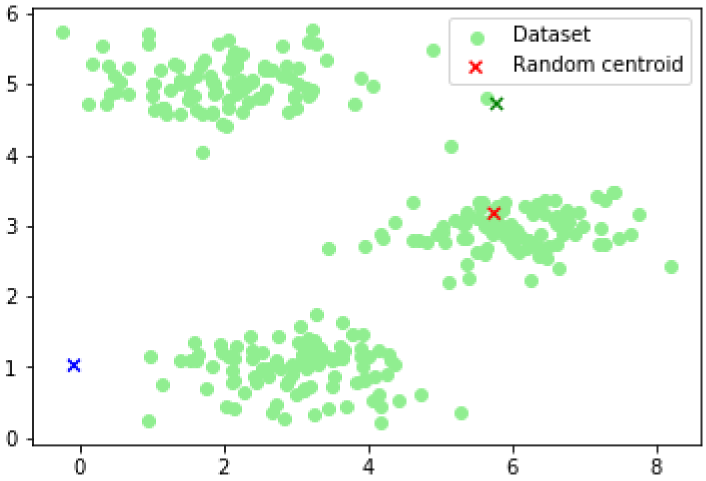
In [ ]:

In [90]:

```
# 1)The initial point distribution should be depicted in the 2-D space, alongs
ide 3 initial centroid points.
#The latter should be in 3 different colors and specific markers indicating th
ey are centroids and not points.
#The dataset points should be in any 4th color. So essentially 2 markers and 4
colors in total. 1 plot submission.
# I have used Random Centroid genertion.
m=[]
m=k_centroid()
m=np.array(m)
plt.scatter(x,y, marker='o',color='lightgreen',label='Dataset')
col1=['red','green','blue']
plt.scatter(m[:,0],m[:,1], marker='x',color=col1, label='Random centroid')
plt.legend(loc=0)
```

Out[90]:

```
<matplotlib.legend.Legend at 0x1e76cac58c8>
```

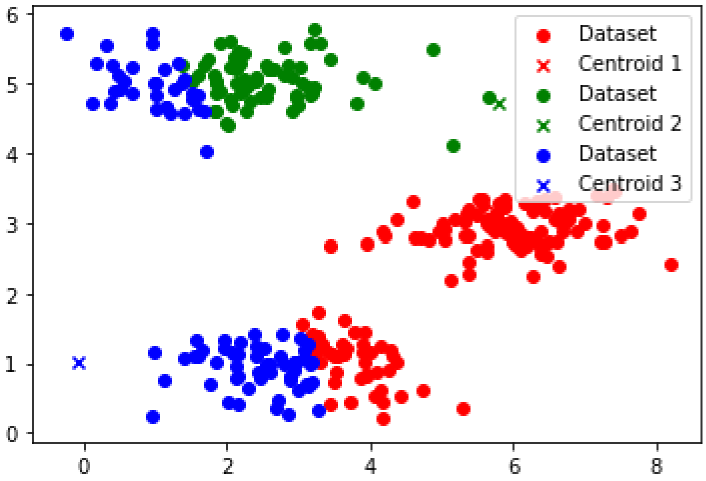


In [91]:

```
#2)     Subsequent to the above step, once the points are assigned to its closest centroid,
#color each point to its assigned centroid color.
#The centroid should still show in same color, but different marker.
#So now essentially 2 markers and 3 colors in total. 1 plot submission.
g = ("Centroid 1", "Centroid 2", "Centroid 3")
col=['orange','lightgreen','lightblue']
newt=classify(t,m)
for k in range(3):
    ra=[]
    ra1=[]
    for i in range(len(newt)):
        if newt[i]==k:
        #        plt.scatter(v[i],y[i],'o', color='black')
            ra.append(x[i])
            ra1.append(y[i])
    plt.scatter(ra,ra1,marker='o',color=col1[k],label='Dataset')
    plt.scatter(m[k,0],m[k,1], marker='x',color=col1[k],label=g[k])
plt.legend(loc=1)
```

Out[91]:

```
<matplotlib.legend.Legend at 0x1e76ca0cb48>
```



In [92]:

```
for l in range(10):
    newt=classify(t,m)
    m=new_cent(t,newt)
    print('iteration',l+1)
    print(m)
#plt.scatter(x,y)
#plt.scatter(m[:,0],m[:,1], marker='x',color='red')
```

```
iteration 1
[[5.31248692 2.34390183]
 [2.58600641 5.0438481 ]
 [1.86893321 2.45030346]]
iteration 2
[[5.59351063 2.52946927]
 [1.98363152 5.03043004]
 [2.64333963 1.01843285]]
iteration 3
[[6.00447483 2.92407086]
 [1.98363152 5.03043004]
 [2.99471436 1.00871148]]
iteration 4
[[6.03366736 3.00052511]
 [1.95399466 5.02557006]
 [3.04367119 1.01541041]]
iteration 5
[[6.03366736 3.00052511]
 [1.95399466 5.02557006]
 [3.04367119 1.01541041]]
iteration 6
[[6.03366736 3.00052511]
 [1.95399466 5.02557006]
 [3.04367119 1.01541041]]
iteration 7
[[6.03366736 3.00052511]
 [1.95399466 5.02557006]
 [3.04367119 1.01541041]]
iteration 8
[[6.03366736 3.00052511]
 [1.95399466 5.02557006]
 [3.04367119 1.01541041]]
iteration 9
[[6.03366736 3.00052511]
 [1.95399466 5.02557006]
 [3.04367119 1.01541041]]
iteration 10
[[6.03366736 3.00052511]
 [1.95399466 5.02557006]
 [3.04367119 1.01541041]]
```

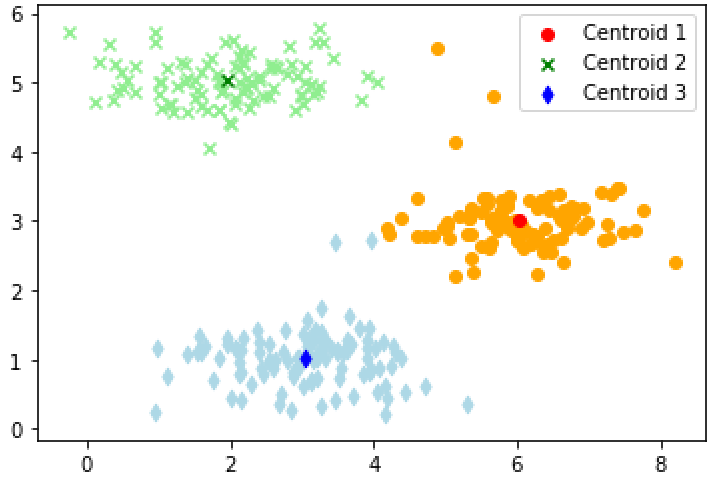
In [93]:

```
#4)     Show the final point distribution, with each point colored according to its final centroid.
#So essentially 2 markers and 3 colors in total. 1 plot submisstion.
# Here I have used 3 different with light and dark background colour for better display
d=['o', 'x', 'd']
col=['orange','lightgreen','lightblue']
g = ("Centroid 1", "Centroid 2", "Centroid 3")
for k in range(3):
    rat=[]
    rat1=[]

    for i in range(len(newt)):
        if newt[i]==k:
    #         plt.scatter(v[i],y[i],'o', color='black')
                rat.append(x[i])
                rat1.append(y[i])
    plt.scatter(rat,rat1, marker=d[k],color=col[k])
    plt.scatter(m[k,0],m[k,1],marker=d[k],color=col1[k], label=g[k])
plt.legend(loc=1)
```

Out[93]:

```
<matplotlib.legend.Legend at 0x1e76cbf3a08>
```



In [ ]: