In [1]:

```
# this guide requires python 3.6
import sys
sys.version
```

Out[1]:

```
'3.6.7 | packaged by conda-forge | (default, Nov  6 2019, 16:19:42) \n[GCC
7.3.0]'
```

In [2]:

```
import numpy as np
import h5py
```

In [3]:

```
# open the file read-only, with error handling
try:
    h5 = h5py.File("pshell-20200824-191714-ManipulatorScan.h5", "r")
except OSError:
    print("error opening file")
```

In [4]:

```
# just checkin'
print(h5.file.filename)
```

```
pshell-20200824-191714-ManipulatorScan.h5
```

In [5]:

```
# what's inside the file at top level?
h5.keys()
```

Out[5]:

```
<KeysViewHDF5 ['__DATA_TYPES__', 'general', 'logs', 'scan 1']>
```

In [6]:

```
# the general group contains the experiment metadata
general = h5['general']
general.keys()
```

Out[6]:

```
<KeysViewHDF5 ['authors', 'pgroup', 'proposal', 'proposer', 'sample']>
```

In [7]:

```
# authors is an array, the other ones are singletons
# call the decode method to convert to string
print([v.decode() for v in general['authors']])
print(general['pgroup'][()].decode())
print(general['sample'][()].decode())
```

```
['Matthias Muntwiler']
p17274
Cu(111)
```

In [8]:

```
# most files contain just one scan group.
# XPS spectra, e.g., can contain multiple scans.
#
# what's inside a scan group?
h5['scan 1'].keys()
```

Out[8]:

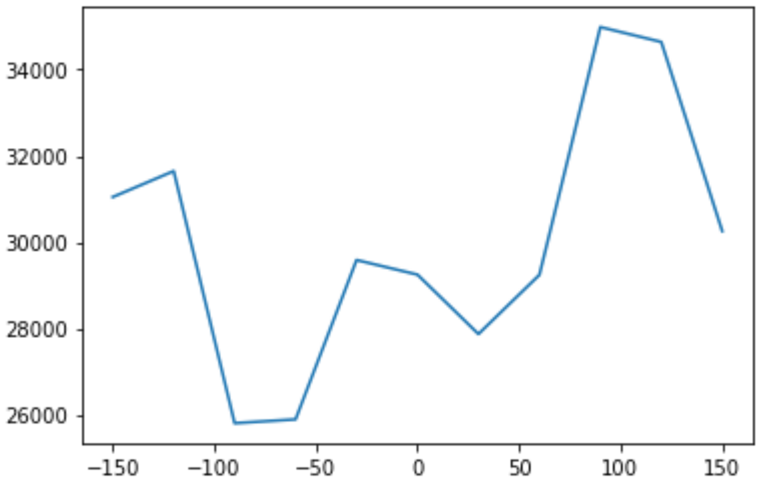
```
<KeysViewHDF5 ['Counts', 'ImageAngleDistribution', 'ImageEnergyDistribution
', 'MachineCurrent', 'ManipulatorPhi', 'RefCurrent', 'SampleCurrent', 'Scien
ta image', 'Scienta spectrum', 'attrs', 'meta']>
```

In [9]:

```
import matplotlib.pyplot as plt
```

In [10]:
```
# the datasets (all items except 'attrs' and 'meta') contain the measured da
ta
# they are represented as numpy datasets and can be plotted directly
plt.plot(h5['scan 1']['ManipulatorPhi'], h5['scan 1']['Counts'])
```

Out[10]:
```
[<matplotlib.lines.Line2D at 0x7ff16cb72a90>]
```



In [11]:
```
# the attrs group contains beamline status
h5['scan 1']['attrs'].keys()
```

Out[11]:
```
<KeysViewHDF5 ['AcquisitionMode', 'AnalyserSlit', 'BeamlinePressure', 'Chamb
erPressure', 'DetectorMode', 'DynamicBumpYAngle', 'DynamicBumpYOffset', 'Ele
mentSet', 'EnergyMode', 'ExcitationEnergy', 'ExitSlit', 'FocusWaterTemp', 'F
ocusXRot', 'FocusYRot', 'FocusYTrans', 'FocusZRot', 'FocusZTrans', 'Frontend
HCenter', 'FrontendHSize', 'FrontendVCenter', 'FrontendVSize', 'LensMode', '
MachineBumpXAngle', 'MachineBumpXOffset', 'MachineBumpYAngle', 'MachineBumpY
Offset', 'MachineCurrent', 'ManipulatorCoolFlow', 'ManipulatorCoolFlowSet',
'ManipulatorPhi', 'ManipulatorTempA', 'ManipulatorTempB', 'ManipulatorTheta
', 'ManipulatorTilt', 'ManipulatorX', 'ManipulatorY', 'ManipulatorZ', 'MonoA
pertureMode', 'MonoBeta', 'MonoBladeDown', 'MonoBladeUp', 'MonoCff', 'MonoEn
ergy', 'MonoGrating', 'MonoHCenter', 'MonoHSize', 'MonoTheta', 'MonoVCenter
', 'MonoVSize', 'NumIterations', 'PassEnergy', 'RefCurrent', 'RefCurrentGain
', 'RefocusHCenter', 'RefocusHSize', 'RefocusVCenter', 'RefocusVSize', 'Refo
cusXRot', 'RefocusYRot', 'RefocusYTrans', 'RefocusZRot', 'RefocusZTrans', 'S
ampleCurrent', 'SampleCurrentGain', 'ScientaCenterEnergy', 'ScientaChannelBe
gin', 'ScientaChannelEnd', 'ScientaDwellTime', 'ScientaHighEnergy', 'Scienta
LowEnergy', 'ScientaNumChannels', 'ScientaNumSlices', 'ScientaSliceBegin', '
ScientaSliceEnd', 'StepSize']>
```

In [12]:
```
# create a shortcut to the first scan group
# scan1 and h5['scan 1'] can be used interchangeably
scan1 = h5['scan 1']
print(scan1)
```

```
<HDF5 group "/scan 1" (11 members)>
```

In [13]:
```
# a scan group also contains direct 'attributes'
# note that 'attributes' is used with several different meanings - unfortuna
tely:
# - the scan group contains another group called 'attrs'
# - the scan group has 'attributes', which are represented in h5py by the 'a
ttrs' attribute :-o
#
scan1.attrs.keys()
```

Out[13]:
```
<KeysViewHDF5 ['Dimensions', 'End', 'Passes', 'PlotTypes', 'Readables', 'Sta
rt', 'Steps', 'Writables']>
```

In [14]:

```python
# number of scan dimensions
print(scan1.attrs['Dimensions'])
# number of scan steps
print(scan1.attrs['Steps'])
# number of iterations (XPS spectra only)
try:
    print(scan1.attrs['Iterations'])
except KeyError:
    print("(no iterations)")
# number of passes
print(scan1.attrs['Passes'])
# start and end time in microseconds
print((scan1.attrs['End'] - scan1.attrs['Start']) / 1000 / 60)
```

```
1
[10]
(no iterations)
1
7.917883333333333
```

In [15]:

```python
# list of scan positioners
scan1.attrs['Writables']
```

Out[15]:

```
array([b'ManipulatorPhi'], dtype='|S14')
```

In [16]:

```python
# remember to cast to string
scan1.attrs['Writables'][0].decode()
```

Out[16]:

```
'ManipulatorPhi'
```

In [17]:

```python
# list of detectors
scan1.attrs['Readables']
```

Out[17]:

```
array([b'Counts', b'MachineCurrent', b'SampleCurrent', b'RefCurrent',
       b'Scienta spectrum', b'ImageEnergyDistribution',
       b'ImageAngleDistribution', b'Scienta image'], dtype='|S23')
```

In [18]:

```python
# note - these are the names of the datasets in the scan group!
scan1.keys()
```

Out[18]:

```
<KeysViewHDF5 ['Counts', 'ImageAngleDistribution', 'ImageEnergyDistribution
', 'MachineCurrent', 'ManipulatorPhi', 'RefCurrent', 'SampleCurrent', 'Scien
ta image', 'Scienta spectrum', 'attrs', 'meta']>
```

In [ ]:

```python

```