Pymatgen_Structures_Example

November 23, 2018

In [20]:
```
# An example of how to set up some electronic and optical calculations using VASP.
# Assumming an INCAR, KPOINTS and POTCAR file have already been made...

# 1. Download a structure from Materials Project or write your own POSCAR from scratch
# For this example we will download the MoS2 bulk compound.
```

In [21]:
```
# Download structure from Materials Project
import pymatgen as mg                          # Imports basic pymatgen funcs.
from pymatgen.ext.matproj import MPRester      # Imports API for MP material database
from pymatgen.io.vasp.inputs import *          # Imports pymatgen/VASP input funcs
import os                                      # Useful Linux/Python library

# Use your Materials Project API key from profile
myAPI = ''  # <-- input you API key here
que = MPRester(myAPI)
```

In [22]:
```
# Download the structures of MoS2 Bulk from the MP database
# MP id - MoS2 = 'mp-2815'
cell_MoS2 = que.get_structure_by_material_id(material_id='mp-2815',
                                             conventional_unit_cell=True,
                                             final=True)

# Notes - conventional_unit_cell = Boolean - True for conventional cell, False for prim
#       - final = Boolean - True if final relaxed structure, False for pre-relaxed stru
```

In [23]:
```
# You can see a summary of the structure data by printing out the structure objs
print(cell_MoS2)
```

```
Full Formula (Mo2 S4)
Reduced Formula: MoS2
abc   :   3.190316   3.190316  14.879004
angles:  90.000000  90.000000 120.000000
Sites (6)
  #  SP           a         b         c
---  ----  --------  --------  --------
  0  Mo    0.666667  0.333333  0.75
  1  Mo    0.333333  0.666667  0.25
  2  S     0.333333  0.666667  0.855174
```

```
  3  S     0.666667  0.333333  0.355174
  4  S     0.333333  0.666667  0.644826
  5  S     0.666667  0.333333  0.144826
```

In [24]:
```
# We can convert the structure into a POSCAR file with pymatgen code:
Poscar(structure=cell_MoS2,
       selective_dynamics=False,
       comment='MP POSCAR: Hexagonal-MoS2 Bulk ').write_file('POSCAR_MoS2_Bulk')
os.listdir()
```

Out[24]:
```
['POSCAR_MoS2_Bulk',
 'POSCAR_MoS2_2D',
 '.ipynb_checkpoints',
 'Pymatgen_Structures_Example.ipynb',
 'poscars',
 'POSCAR_2d_MoS2',
 'Band_exmp.ipynb']
```

In [25]:
```
# See the Word doc to see how to write make a POSCAR manually.
# You can make a POSCAR for the 2D a few ways...
```

In [26]:
```
# You could the sites of one of the layers...

# Lets remove these sites:
print(cell_MoS2[0],
      cell_MoS2[2],
      cell_MoS2[4])

# To remove use the remove_sites() method
cell_MoS2.remove_sites([0,2,4])

print(cell_MoS2)
```

```
[ 1.59515786 -0.92096482 11.15925323] Mo [ 1.59515786  0.92096482 12.72414238] S [1.59515786 0.9
Full Formula (Mo1 S2)
Reduced Formula: MoS2
abc   :   3.190316   3.190316  14.879004
angles:  90.000000  90.000000 120.000000
Sites (3)
  #  SP           a         b         c
---  ----  --------  --------  --------
  0  Mo    0.333333  0.666667  0.25
  1  S     0.666667  0.333333  0.355174
  2  S     0.666667  0.333333  0.144826
```

In [27]:
```
# Translate the sites so that Mo is centered in the c direction
cell_MoS2.translate_sites(indices=[0,1,2],vector=[0,0,0.25])  # shift atoms by 0.25 fra
print(cell_MoS2)
```

```
Full Formula (Mo1 S2)
Reduced Formula: MoS2
abc   :   3.190316   3.190316  14.879004
angles:  90.000000  90.000000 120.000000
Sites (3)
  #  SP           a         b         c
---  ----  --------  --------  --------
  0  Mo    0.333333  0.666667  0.5
  1  S     0.666667  0.333333  0.605174
  2  S     0.666667  0.333333  0.394826
```

In [28]:
```
print(cell_MoS2)
```

```
Full Formula (Mo1 S2)
Reduced Formula: MoS2
abc   :   3.190316   3.190316  14.879004
angles:  90.000000  90.000000 120.000000
Sites (3)
  #  SP           a         b         c
---  ----  --------  --------  --------
  0  Mo    0.333333  0.666667  0.5
  1  S     0.666667  0.333333  0.605174
  2  S     0.666667  0.333333  0.394826
```

In [29]:
```
# Or we can build one from scratch using pymatgen.
# To do so, you need to define a lattice matrix, the species, and their positions.

# First define a lattice matrix using the pymatgen.Lattice library:
latt = mg.Lattice.from_lengths_and_angles(abc=[3.190316,3.190316, 14.879004],
                                          ang=[90.0,90.0,120.0])


# Define the atomic species:
spec_list = ['Mo','S','S']

# Define the atomic positions in an array
#atom_pos = np.array([[0.33333, 0.66667, 0.50000],[0.66667,0.33333,0.605174],[0.66667,0
atom_pos = np.array([[1/3,2/3,1/2],[2/3,1/3,0.605174],[2/3,1/3,0.394826]])


# Use the pymatgen.Structure function to create a structure object
cell_2d = mg.Structure(lattice=latt,
                       species=spec_list,
                       coords=atom_pos,
                       coords_are_cartesian=False,)
print(cell_2d)
```

```
Full Formula (Mo1 S2)
Reduced Formula: MoS2
abc   :   3.190316   3.190316  14.879004
angles:  90.000000  90.000000 120.000000
Sites (3)
  #  SP           a         b         c
---  ----  --------  --------  --------
  0  Mo    0.333333  0.666667  0.5
  1  S     0.666667  0.333333  0.605174
  2  S     0.666667  0.333333  0.394826
```

In [30]:
```
# Write to POSCAR file
Poscar(structure=cell_2d,comment='2D Hexagonal MoS2').write_file('POSCAR_MoS2_2D')
os.listdir()
```

Out[30]:
```
['POSCAR_MoS2_Bulk',
 'POSCAR_MoS2_2D',
 '.ipynb_checkpoints',
 'Pymatgen_Structures_Example.ipynb',
 'poscars',
 'POSCAR_2d_MoS2',
 'Band_exmp.ipynb']
```

In [ ]: