Creating an Image Catalog

- GDAL and OGR
- read dimensions and georeferencing of raster data
- create a shapefile with new fields
- write features to shapefile
- glob files using glob module
- get file info using os.path module

os.path

```
>>> import os.path
>>> os.path.abspath('./hobu.txt')
'P:\\OSG05\\aggregation\\hobu.txt'
>>>
>>> os.path.basename('P:/OSG05/aggregation/hobu.txt')
'hobu.txt'
>>>
>>> os.path.getctime('hobu.txt')
1118386365
>>>
```

glob

```
>>> import glob
>>> glob.glob('*.txt')
['hobu.txt']
>>>
```

glob + os.path

```
>>> for path in glob.glob('*.txt'):
...     print os.path.basename(path), \
...           os.path.abspath(path), \
...           os.path.getctime(path)
...
hobu.txt P:\\OSG05\\aggregation\\hobu.txt 1118386365
>>>
```

gdal

```
>>> import gdal
>>> dataset = gdal.Open('P:/OSG05/cameron30_zip.tif')
>>> dataset
<gdal.gdal.Dataset instance at 0x008E48C8>
>>>
```

GDAL Dataset Properties

```
>>> dataset.RasterCount
3
>>> dataset.RasterXSize
999
>>> dataset.RasterYSize
1586
>>> dataset.GetGeoTransform()
(-106.05969999999999, 0.00027777777777799998, 0.0,
40.842500000000001, 0.0, -0.00027769230769199998)
>>>
```

Dataset Properties 2

```
>>> for path in glob.glob('P:/OSG05/*.tif'):
...     ds = gdal.Open(path)
...     geo = ds.GetGeoTransform()
...     pixels = ds.RasterXSize
...     lines = ds.RasterYSize
...     minx = geo[0]
...     maxx = minx + pixels * geo[1]
...     maxy = geo[3]
...     miny = maxy + lines * geo[5]
...     print os.path.basename(path), \
              (minx, miny, maxx, maxy)
...
escalante30_zip.tif (-111.705, 37.68638888056207,
-111.22944443999971, 38.06583333055551)
```

Create Shapefile with ogr

```
>>> import ogr
>>> driver = ogr.GetDriverByName('ESRI Shapefile')
>>> shp = driver.CreateDataSource('example.shp')
>>> layer = shp.CreateLayer('example',
                            geom_type=ogr.wkbPolygon)
>>> shp.Destroy()
>>>
```

Produces a shapefile with no features

Create Image Catalog

```
>>> shp = driver.CreateDataSource('catalog.shp')
>>> catalog = shp.CreateLayer('catalog',
                              geom_type=ogr.wkbPolygon)
```

Add Fields

A string field named 'abspath' to hold filesystem path to datasets

```
>>> field = ogr.FieldDefn('abspath', ogr.OFTString)
>>> field.SetWidth(200)
>>> catalog.CreateField(field)
0
```

A field named 'mtime' to hold the modification time of datasets

```
>>> field = ogr.FieldDefn('mtime', ogr.OFTInteger)
>>> field.SetWidth(12)
>>> catalog.CreateField(field)
1
```

Adding a Feature

```
>>> filename = 'escalante30_zip.tif'
>>> feature = ogr.Feature(catalog.GetLayerDefn())
>>> feature.SetField(0, os.path.abspath(filename))
>>> feature.SetField(1, os.path.getmtime(filename))
```

Feature Geometry

Make WKT representation of previous dataset bounds

```
>>> bounds = [-111.705, 37.686388888056207,
              -111.22944443999971, 38.065833333055551]
>>> t = 'POLYGON ((%(x0)f %(y0)f, %(x0)f %(y1)f, %(x1)f %(y1)
f, %(x1)f %(y1)f, %(x0)f %(y0)f))'
>>> wkt = t % {'x0': bounds[0], 'y0': bounds[1],
               'x1': bounds[2], 'y1': bounds[3]}
```

Make an ogr.Geometry and set the feature's geometry

```
>>> geom = ogr.CreateGeometryFromWkt(wkt)
>>> feature.SetGeometryDirectly(geom)
0
```

Finishing the Catalog

Create a new feature in our layer based upon this one, and close the data source.

```
>>> catalog.CreateFeature(feature)
0
>>> shp.Destroy()
```

Produces a shapefile with a dataset extent as the single feature.

Onward

- glob is only the start, see os.walk()
- float fields
- many other vector formats
- many other raster formats
- read/write access to raster band data
- raster reprojection