Introduction to Python

Feature Summary, Session 6

READING MULTIDIMENSIONAL STRUCTURES

- **pprint.pprint()** function: ***print a structure in readable format***

```
import pprint
pprint.pprint(messyobj)
```

- list of lists: ***a list in which each item is a reference to a list***

```
x = [ [1, 2, 3],
      [4, 5, 6]  ]
```

- list of dicts: ***a list in which each item is a reference to a dict***

```
x = [{'id':45, 'name':'Jo'},
     {'id':46, 'name':'Bo'}]
```

- dict of lists: ***a dict in which each item is a reference to a list***

```
x = [ 45: [3, 9, 17],
      46: [2, 7, 22]  ]
```

- dict of dicts: ***a dict in which each item is a reference to a dict***

```
x = {45: {'fname': 'Joe',
          'lname': 'Jones'},
     46: {'fname': 'Max',
          'lname': 'Marx'} }
```

arbitrary structure: ***any container containing references to other containers, to any depth***
(at right: a list of dicts of dict/lists)

```
x = [{'a': {'x': 5,
            'y': 10},
      'b': [1, 2, 3] },
     {'a': {'x': 7,
            'y': 3},
      'b': [4, 5, 6] } ]
```

BUILDING MULTIDIMENSIONAL STRUCTURES

- build a list of lists from a file
  (note: example at right is generic and not meant to apply to any specific situation)

```
lol = []
for line in open('fi.txt'):
    lol.append(line.split())
```

- build a list of dicts from a fiile
  (note: example at right is generic and not meant to apply to any specific situation)

```
lod = []
 for line in open('fi.txt'):
    items = line.split()
    d = {'a': items[0],
         'b': items[1]}
    lod.append(d)
```

- build a dict of dicts from a file
  (note: example at right is generic and not meant to apply to any specific situation)

```
dol = {}
for line in open('fi.txt'):
    items = line.split()
    key = items[0]
    d[key] =
        { 'a': items[1],
          'b': items[2]  }
```

- accumulate a dict of lists from a file
  (note: example at right is generic and not meant to apply to any specific situation)

```
dol = {}
for line in open('fi.txt'):
    items = line.split()
    key = items[0]
    if key not in d:
        d[key] = 0
    d[key] +=float(items[1])
```