

Reading and In-Class Problem

Reading

- Langtangen, Primer on Scientific Programming with Python: Ch. 4

Problem

- Using the provided data set
 - Plot the temperature and water levels over time
 - What is the maximum, average, and minimum temperature recorded at this site?
 - Calculate and plot a daily running average for temperature.

Numpy

- array processing environment
- new 'array' data structure
- faster
- access with numpy module
- more convenient indexing?

```
import numpy as np
A=[[1,2],[11,12]]
A=np.array(A)
```

Making Arrays

- Arrays have uniform numeric data type
- same number of elements in rows, columns, etc.
- making array
 - make list and convert
 - make array using array operations
 - make array and fill with values

```
import numpy as np

#make array filled with zeros and ones
A=np.zeros([3,3])
A=np.zeros([3,3],dtype=int)
A=np.ones([4,3])

#overwrite values by looping
for i in range(4):
    for j in range(3):
        A[i,j]=i+j

#making list and converting
A=[[i+j for j in range(3)] for i in range(4)]
A=np.array(A)
```

Manipulating arrays

```
import numpy as np
A=np.arange(6)
A.reshape([3,3]) #error!
A.reshape([3,2])
np.reshape(A,[3,2])# same thing
A=np.arange(36)
# figures out missing dimension automagically
A=np.reshape(A,[6,-1])
A.shape
np.shape(A)
A[2:4,4:6]#slicing
A[:,3,::3]#strides
```

- Element-wise operations
- vectorized data

```
import numpy as np

def calculate(a):
    b=a*10+17
    return b

A=np.arange(16)
A=A.reshape([4,4])
B=A.copy() #why do we do this?
C=np.sqrt(B)
C=B*5.
C=B+10.
C=B**2-C

C=calculate(10)
C=calculate(B)
```

Picking apart arrays

```
import numpy as np

A=np.arange(25)
A=A.reshape([5,5])*3.
np.take(A,[1,11,15])#select based on (continuous) index
A.take([1,3,4],axis=0) #row selection
A.take(1,3,4,axis=1) #column selection
np.where((A>2))#ID indexes
np.where((A>2)&(A<5))#numpy 'and'
np.where((A>2)|(A==0))#numpy 'or'
np.where(A>2,1,0)#put true/false values in positions
np.put(A,[0,11],[100,101])#overwrites value in place
```

Matplotlib

- excellent plotting library
- add on modules for map projections (basemap) and 3-D plots (mplot3d)

- `ipython -pylab -wthread`

```
import numpy as np
import pylab as pl
x=np.arange(100)
x=x*4*np.pi/100.

y=np.sin(x)
z=np.cos(x)

pl.plot(x,y)
pl.plot(x,z)
pl.show()
```

Matplotlib

```
import numpy as np
import pylab as pl
x=np.arange(100)
x=x*4*np.pi/100.

y=np.sin(x)
z=np.cos(x)

pl.figure(1,figsize=(6,6))
pl.subplot(211)
pl.plot(x,y)
pl.subplot(212)
pl.plot(x,z,linestyle='None',marker='o',color='red')
pl.show()
```

dateutil

- 'easy' date manipulation
- converts text date into datetime object
- `parser.parse('2011-01-01 10:10:10')`
- useful for plotting value vs. date-time