

8 Functions

```
def name_of_function(arg1,arg2):
```

8.1 Calling a function

```
x=5
y=7
z=max(x,y) #max is the function. x and y are the arguments
print(z) #print is the function. z is the argument
7
```

8.2 Importing functions

```
import math #imports the math namespace into our program namespace
math.sqrt(34.23) #Functions in the math namespace have to be disambiguated
```

```
import math as m #imports the math namespace into our program namespace
but gives it the name 'm'
m.sqrt(34.23) #Functions in the math namespace have to be disambiguated
using the name 'm' rather than 'math'
```

A Few Useful Functions

All of the functions come from the "`numpy`" package.

- `np.mean` - For calculating a mean
- `np.std` - For calculating standard deviation
- `np.var` - For calculating variance
- `np.sqrt` - For taking the square root of a number (instead of `x \** .5`)
- `np.min` - For finding the minimum value of a collection
- `np.max` - For finding the maximum value of a collection

9 Python Libraries

9.1 Numpy

Creating a numpy array

```
import numpy as np
ax = np.array([1,2,3,4,5])
print(x,id(x),len(x))
```

Specifying the type

Useful when reading a text stream directly into a numerical array

```
x=['1','2','3']
xi = np.array(x,'int')
xf = np.array(x,'float')
xs = np.array(x,'str')
print(xi,xf,xs,sep='\n')
```

Basic operations

```
x = np.array([13,24,21.2,17.6,21.7],'float')
print(x.sum(),x.mean(),x.std(),sep='\n')
```

Indexing

```
ax[1,3] #indexing
```

Slicing

```
ax[1:3,2:4]
#ax[:,2:]
```

Matrix multiplication

```
ax = np.arange(10)
ay = np.array([ax,ax])
#Scalar multiplication
ay*2
```

Selecting elements from an np array

```
x=[[0,1,2,3,4,5],[10,11,12,13,14,15],[20,21,22,23,24,25]]
ax=np.array(x,float)
np.where(ax%2==0,1,0)
```

#linalg, a linear algebra module

#functions dealing with polynomials, differentials, etc

```
import scipy
scipy.nanmean(x)
```

9.2 Pandas

#installing pandas libraries

```
!pip install pandas-datareader
```

```
!pip install --upgrade html5lib==1.0b8
```

#There is a bug in the latest version of html5lib so install an earlier version

#Restart kernel after installing html5lib

Imports

```
import pandas as pd #pandas library
```

```
from pandas_datareader import data #data readers (google, html, etc.)
```

#The following line ensures that graphs are rendered in the notebook

```
%matplotlib inline
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt #Plotting library
```

```
import datetime as dt #datetime for timeseries support
```

The structure of a dataframe

```
pd.DataFrame([[1,2,3],[1,2,3]],columns=['A','B','C'])
```

	A	B	C
0	1	2	3
1	1	2	3

Accessing columns and rows

```
df = pd.DataFrame([['r1','00','01','02'],['r2','10','11','12'],['r3','20','21','22']],columns=['row_label','A','B','C'])
```

```
print(id(df))
```

```
df.set_index('row_label',inplace=True)
```

```
print(id(df))
```

```
df
```

Getting column data

```
df['B']
```

Getting row data

```
df.loc['r1']
```

Getting a row by row number

```
df.iloc[0]
```

Getting multiple columns

```
df[['B','A']] #Note that the column identifiers are in a list
```

Getting a specific cell

```
df.loc['r2','B']
```

```
df.loc['r2']['A']
```

Slicing

```
df.loc['r1':'r2']
```

```
df.loc['r1':'r2','B':'C']
```