Quick guide on the Matrix class and the Numpy package

1: The Matrix Class

The Matrix class is just a decorator class for the numpy arrays. You can give it an list, or a list of lists, and it makes a Matrix. In our case, you can give it the filename of a phylip distance matrix, and it reads in the matrix.

```
D = Matrix(filename)
print D
[ 0.00 0.23 0.16 0.20 0.17 ]
[ 0.00 0.23 0.17 0.24 ]
[ 0.00 0.20 0.11 ]
[ 0.00 0.21 ]
[ 0.00 ]
```

When you iterate over the Matrix, it only iterates over half of the matrix, and returns a tuple structured like (i, j, value), so you know which entry it corresponds to in the matrix, and value stored there.

```
for i in D:
    print i
(0, 1, 0.23000000000000001)
(0, 2, 0.16)
(0, 3, 0.20000000000000001)
(0, 4, 0.17000000000000001)
(1, 2, 0.23000000000000001)
(1, 3, 0.17000000000000001)
(1, 4, 0.23999999999999999)
(2, 3, 0.20000000000000001)
```

```
(2, 4, 0.11)
(3, 4, 0.20999999999999999)
```

It also does not include the entries where we compare (i,i) because these would always be 0, and wouldn't be informative.

Since the iterator returns a tuple, you can unpack values, into the desired values like this:

```
for i, j, value in D:
    print i, j, value
0 1 0.23
0 2 0.16
0 3 0.2
0 4 0.17
1 2 0.23
1 3 0.17
1 4 0.24
2 3 0.2
2 4 0.11
3 4 0.21
```

When the Matrix iterates over the data, it uses an internal variable, called indices, which is the list of indices to access. Then it also contains the names of the leaves from the read-in data called names.

```
D.indices
[0, 1, 2, 3, 4]
D.names
['A', 'B', 'C', 'D', 'E']
```

The Matrix class also contains a function called removeIndex(i), where it deletes the specified index i from the indices list.

```
print D
[ 0.00 0.23 0.16 0.20 0.17 ]
[ 0.00 0.23 0.17 0.24 ]
[ 0.00 0.20 0.11 ]
[ 0.00 0.21 ]
[ 0.00 ]
D.removeIndex(2)
print D
[ 0.00 0.23 0.20 0.17 ]
[ 0.00 0.17 0.24 ]
[ 0.00 0.21 ]
[ 0.00 ]
```

2: Numpy

The Matrix class inherits all of the properties that numpy arrays contain. So any functionality assigned to numpy arrays applies to the Matrix class.

Typically people import numpy as the name np, so you can use write np.sum() instead of numpy.sum()

```
import numpy as np
```

When you have numpy matrix, you access values, and can assign values using [i,j].

```
D[i,j]
D[i,j] = value
```

If you want the data from an entire row or column, you would simply leave one of the values empty. You can also of course use slicing like [1:4,2:3]. If you want to do this to the Matrix class use the following function:

```
np.asarray(D[1:4,2:3])
```

The matrix class does do this, but if you attempt to print it, there is no representation of it, therefore converting it back into a standard np.array would make it presentable.

Another useful feature of Numpy for the Matrix Class, is that you can ask for the size or shape of the matrix.

```
D.shape
(5,5)
```

Regardless of whether the indices list changes, the shape will remain the same.