Homework 2: NumPy Arrays and Matrices

Objectives:
- Learn about NumPy arrays and matrices and the differences between them.

NumPy Operations

NumPy supports multi-dimensional array called ndarray. In NumPy, a matrix is a special 2D array that allows matrix multiplication. In most cases, a 2D array can be regarded as the same as a matrix, except for matrix multiplication.

1. Run Python 2.7 IDLE (Integrated Development Environment).

2. Import NumPy
   ```
   >>> import numpy as np
   ```

3. 1D arrays
   - Create 1D arrays.
     ```
     >>> a = np.array([1, 2, 3])
     >>> a
     >>> b = np.array([0.1, 0.2, 0.3])
     ```
   - Array elements are indexed from 0 to n-1, where n is the number of elements.
     ```
     >>> for i in range(len(b)):
     ...     print i, b[i]
     ```

   Notes:
   - The print statement should be indented by a tab character.
   - After typing the print statement, press the enter key two times to execute.

   - Operations on arrays are performed in an element-by-element manner.
     ```
     >>> 0.5 + a
     >>> 0.5 * a
     >>> a + b
     >>> a - b
     >>> a * b
     >>> a / b
     ```
Explicit loop over array elements runs slower, particularly for large arrays.

```python
>>> c = np.zeros(3)
>>> for i in range(len(c)):
    c[i] = a[i] + b[i]
>>> print c
```

So, it is advisable to use the built-in array operations that not only runs faster but are also more compact and easier to read.

4. 2D arrays
- Create 2D arrays.

```python
>>> a = np.array([[1, 2, 3], [4, 5, 6]])
>>> print a
>>> b = np.array([[0.1, 0.2, 0.3], [0.4, 0.5, 0.6]])
>>> print b
```

- Get an element.

```python
>>> a[1, 2]
```

- Get row 0, which returns an array of one row.

```python
>>> a[0, :]
```

- Get column 1, which also returns an array of one row.

```python
>>> a[:, 1]
```

- Operations on arrays are performed in an element-by-element manner.

```python
>>> 0.5 + a
>>> 0.5 * a
>>> a + b
>>> a - b
>>> a * b
>>> a / b
```

5. Matrices
- Create matrices.

```python
>>> a = np.matrix([[1, 2, 3], [4, 5, 6]])
>>> print a
>>> b = np.matrix([[0.1, 0.2, 0.3], [0.4, 0.5, 0.6]])
>>> print b
```
- Get an element.
  
  >>> a[1,2]

- Get row 0, which returns a row matrix.
  
  >>> a[0,:]

- Get column 1, which returns a column matrix. This operation is different from that for 2D arrays.
  
  >>> a[:,1]

- These matrix operations produce the same results as those for 2D arrays.
  
  >>> 0.5 + a
  >>> 0.5 * a
  >>> a + b
  >>> a - b
  >>> a / b

- This operation is invalid because the two matrices are not of compatible sizes.
  
  >>> a * b

- Matrix multiplication is valid only if the matrices have compatible sizes.
  
  >>> c = np.array([[0.1, 0.2], [0.3, 0.4], [0.5, 0.6]])
  >>> print a.shape, c.shape
  >>> a * c

6. Special arrays

- Create special 2D arrays.
  
  >>> np.empty([2,3])  # array with arbitrary values
  >>> np.eye(2,3)  # with ones along diagonal and zeros elsewhere
  >>> np.identity(3)  # identity array
  >>> np.ones([2,3])  # array filled with ones
  >>> np.zeros([2,3])  # array filled with zeros
  >>> np.random.rand(2,3)  # array with random values

- Convert 2D arrays into matrices.
  
  >>> a = np.zeros([2,3])
  >>> a
  >>> b = np.matrix(a)
  >>> b
  >>> c = np.matrix(np.zeros([2,3]))  # short-hand method
  >>> c
7. Adding, subtracting or dividing array with matrix results in a matrix.

```python
>>> a = np.matrix(np.eye(2,3))
>>> a
>>> b = np.ones([2,3])
>>> b
>>> a - b
>>> b - a
>>> a / b
>>> b / a
```

Note: Mixing arrays and matrices in an expression can create confusion. It is advisable not to mix arrays and matrices in an expression.

**Summary**

Note the similarities and differences between array and matrix operations.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Arrays</th>
<th>Matrices</th>
<th>Similar?</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>element-by-element addition</td>
<td>element-by-element addition</td>
<td>yes</td>
</tr>
<tr>
<td>-</td>
<td>element-by-element subtraction</td>
<td>element-by-element subtraction</td>
<td>yes</td>
</tr>
<tr>
<td>*</td>
<td>element-by-element multiplication</td>
<td>matrix multiplication</td>
<td>no</td>
</tr>
<tr>
<td>/</td>
<td>element-by-element division</td>
<td>element-by-element division</td>
<td>yes</td>
</tr>
<tr>
<td>[i,:]</td>
<td>returns a row</td>
<td>returns a row</td>
<td>yes</td>
</tr>
<tr>
<td>[:,j]</td>
<td>returns a row</td>
<td>returns a column</td>
<td>no</td>
</tr>
</tbody>
</table>