Sheet 01 answer.

Ahmed Ashraf Mohamed

1 How can you create a Numpy array? What is the shape of a Numpy array and how can you get it?

- using the function `numpy.array(object,data_type)`
- shape is a numpy attribute of the ndarray, when calling the shape attribute it returns a tuple with corresponding number of dimensions

2 How can you perform element-wise operations on Numpy arrays (list the possible element-wise operations)?

- `np.add()` (Addition)
- `np.subtract()` (subtraction)
- `np.multiply()` (multiplication)
- `np.divide()` (division)
- `np.power()` (First array elements raised to powers from second array)
- `np.remainder()` (remainder of division)
- `np.reciprocal()` (reciprocal of the argument)
- `np.abs()` (absolute value)
- `np.floor()` (Floor)
- `np.ceil()` (Ceiling)
- `np.round()` (Round the given number)

3 How to sum all elements of one array ?

```
numpy.sum(array)
```

4 How to transpose a Numpy array?

```
array = numpy.transpose(array)
# or
```

```
array = array.T
```

5 What transpose do to the Numpy array?

Switches the Rows with columns

6 Having matrixA and matrixB both with shape (3,2) how to create a numpy array with shape (2,3,2) using `np.concatenate`?

```
matrixC = np.concatenate(([matrixA],[matrixB]))
```

7 What is the difference between reshape and resize in Numpy?

`reshape()` doesn't modify the data of the numpy array, but `resize()` does change it.

8 How can you extract the diagonal elements of a Numpy array?

```
diagonalElements = np.diag(array)
```

9 What is the difference between `np.dot` and `np.matmul` in Numpy?

- `np.matmul()` is Matrix product of two arrays.
- `np.dot ()` is a dot product of two arrays. The two arrays must be of same size and the output is the same size

10 How can you perform element-wise comparison of two Numpy arrays (list the possible element-wise comparisons)?

- `np.fmin(arr1, arr2)` (Compare minimum)
- `numpy.logical_and(arr1, arr2)` (Compute the truth value of `arr1` AND `arr2`)
- `numpy.logical_or(arr1, arr2)` (Compute the truth value of `arr1` OR `arr2`)

- `numpy.logical_not(arr1, arr2)` (Compute the truth value of `arr1` NOT `arr2`)
- `numpy.logical_xor(arr1, arr2)` (Compute the truth value of `arr1` XOR `arr2`)
- `numpy.greater(arr1,arr2)` (Return the truth value of $(x_1 > x_2)$)
- `numpy.greater_equal(arr1,arr2)` (Return the truth value of $(x_1 \geq x_2)$)
- `numpy.less(arr1,arr2)` (Return the truth value of $(x_1 < x_2)$)
- `numpy.less_equal(arr1,arr2)` (Return the truth value of $(x_1 \leq x_2)$)
- `numpy.equal(arr1,arr2)` (Return the truth value of $(x_1 = x_2)$)
- `numpy.not_equal(arr1,arr2)` (Return the truth value of $(x_1 \neq x_2)$)