

COMP20270

OOP in Python

Assignment 2

MyArray

Due 5th December.

Objective

The objective for this assignment is to create an array class **MyArray** that implements some of the functionality provided by the **numpy ndarray** class. Here is an example of a simply **numpy** array constructed from nested lists. This example shows **min**, **max** and **mean** methods and data access using square brackets notation. **np.array** is a special function for constructing instances of the **ndarray** class, it is not the constructor.

```
import numpy as np
a2 = np.array([[7,2,8],[3,4,5]])
a2
Out[190]:
array([[7, 2, 8],
       [3, 4, 5]])

In [191]:
type(a2)
Out[191]:
numpy.ndarray

In [186]:
a2.max()
Out[186]:
8

In [183]:
a2.min(axis = 1)
Out[183]:
array([2, 3])

In [192]:
a2.mean()
Out[192]:
4.833333333333333

In [189]:
a2[1,2] = 9
a2
Out[189]:
array([[7, 2, 8],
       [3, 4, 9]])
```

Requirements

Construct a class **MyArray** that meets the following requirements.

1. The constructor allows instances to be created from nested lists. It can handle 1D and 2D arrays.

```
ma1 = MyArray([1,2])
ma2 = MyArray([[1,2,3,4,5],[4,5,6,7,8]])
```

2. How you store the data (array elements) internally is up to you, it could be stored as lists or as a dictionary. Discuss your design decisions in Markdown. Just don't store things in a numpy array!

3. Implement **min**, **max** and **mean** methods similar to those in numpy, i.e. if no parameter is passed a single value for the array is returned, if the axis is specified a list of results for that axis is returned.

4. Implement a copy method that will produce a copy of the array.

```
ma2 = MyArray([[1,2,3,4,5],[4,5,6,7,8]])
ma2c = ma2.copy()
ma2d = ma2
ma2[1,1] = 9
ma2d
```

```
Out[412]:
| 1 2 3 4 5 |
| 4 9 6 7 8 |
```

```
In [413]:
ma2c
```

```
Out[413]:
| 1 2 3 4 5 |
| 4 5 6 7 8 |
```

5. Develop a class method **zeros** that will create an array of a specified size where all entries are 0 (i.e. an alternative constructor). It should handle 1D and 2D arrays as follows.

```
za = MyArray.zeros(3,7)
za
```

```
Out[421]:
| 0 0 0 0 0 0 0 |
| 0 0 0 0 0 0 0 |
| 0 0 0 0 0 0 0 |
```

```
In [422]:
z1 = MyArray.zeros(5)
z1
```

```
Out[422]:
| 0 0 0 0 0 |
```

6. Finally, develop `__getitem__`, `__setitem__` and `__repr__` methods so that array elements can be accessed using the square bracket notation and arrays can provide a representation of the contents.

```
ma2
Out[431]:
| 1 2 3 8 5 |
| 4 9 6 7 8 |
In [433]:
ma2[1,3]
Out[433]:
7
In [434]:
ma2[1,3] = 0
ma2
Out[434]:
| 1 2 3 8 5 |
| 4 9 6 0 8 |
```

use a repr here to return a particular representation of an item

Other Requirements

1. The two constructor options should include some error checking to ensure that the data provided is consistent and can be constructed into an array.

Submission: This is an individual (not group) project. Submission is through the Brightspace page. Your submission should comprise your notebook only. Clear all outputs in the notebook before saving for submission. You should use Markdown cells in the notebook to explain any design decisions you have made.