LECTURE 25

NUMPY

MCS 275 Spring 2021

Emily Dumas

LECTURE 25: NUMPY

Course bulletins:

- Thurs discussion students: Worksheet 9 problem 1.
- Project 3 due 6:00pm CDT on Friday March 19. (That's local time in Chicago.)

PRIVATE ATTRIBUTES

Project 3 asks you to make a class `Chroma` that stores a chromaticity, and to not access any of its attributes directly.

```
c1 = Chroma(75,85,95)
c2 = Chroma(10,10,235)
if c1.r > c2.r:   # DON'T DO THIS!
    print("c1 is more red")
```

PRIVATE ATTRIBUTES

Project 3 asks you to make a class `Chroma` that stores a chromaticity, and to not access any of its attributes directly.

```
c1 = Chroma(75,85,95)
c2 = Chroma(10,10,235)
if c1.more_red_than(c2):   # Do this instead
    print("c1 is more red")
```

PRIVATE ATTRIBUTES

Project 3 asks you to make a class `Chroma` that stores a chromaticity, and to not access any of its attributes directly.

```
c1 = Chroma(75,85,95)
c2 = Chroma(10,10,235)
if (c1.r,c1.g,c1.b) == (c2.r,c2.g,c2.b):  # DON'T DO THIS!
    print("c1 equals c2")
```

PRIVATE ATTRIBUTES

Project 3 asks you to make a class `Chroma` that stores a chromaticity, and to not access any of its attributes directly.

```python
c1 = Chroma(75,85,95)
c2 = Chroma(10,10,235)
if c1 == c2:  # Do this instead
    print("c1 equals c2")
```

Many languages (e.g. C++, Java, C#) have mechanisms to enforce this kind of **implementation hiding**. Python does not.

One common convention is to start method and variable names with a single underscore _ if they are intended to be private. (Project 3 doesn't ask you to do that.)

WHEN TO USE PILLOW

Pillow is obviously a good choice if you are making a program that needs image I/O.

Python+Pillow is a good choice for tasks involving batch operations mixed with some logic, e.g.

> *Take these 10,000 images, ignore the ones that are corrupted or blank, resize the others, convert to JPEG, and give them sequential integer names like `00081.jpg`.*

INSTALLING NUMPY

In most cases, pip is all you need:

```
python3 -m pip install numpy
```

Other methods are described in the Numpy docs.

Test:

```
        >>> import numpy
        >>> numpy.__version__
        '1.17.4'
```

IMPORT AS

You can give a module a new name at import time, e.g.

```python
import math as sun
sun.tan(0.5)
```

Since numpy has a lot of global names, some of which appear frequently in code, most people import it with

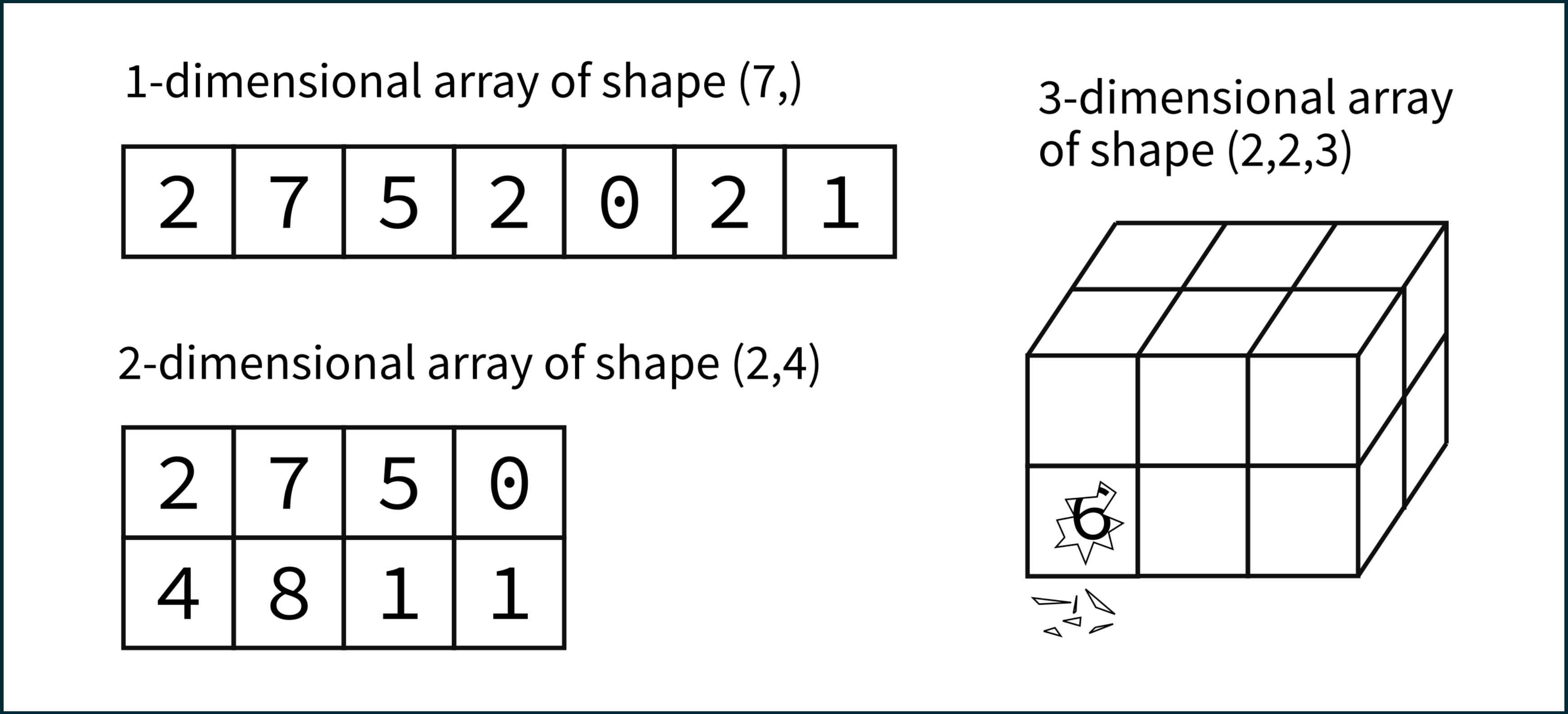
```python
import numpy as np
```

NUMPY PURPOSE

- Fast, statically typed, multidimensional arrays
- Large library of mathematical functions and algorithms (especially linear algebra)

Numpy is one of the most-used Python packages in scientific computing (computational math, data science, machine learning, ...).

ARRAYS


1-dimensional array of shape (7,)
2 7 5 2 0 2 1
2-dimensional array of shape (2,4)
2 7 5 0
4 8 1 1
3-dimensional array of shape (2,2,3)
6

Implemented in `np.ndarray` class.

Without numpy:

```
v = [2,3]
w = [3,-2]
v + w     # [2,3,3,-2]
3*v       # [2,3,2,3,2,3]
v.dot(w)  # fail!
A = [ [2,1], [1,1] ]
type(A)   # list
A*v       # fail!
```

With numpy:

```
v = np.array([2,3])
w = np.array([3,-2])
v + w    # [5,1]
3*v      # [6,9]
v.dot(w) # 0
A = np.array([ [2,1], [1,1] ])
A*v      # possibly confusing answer
A.dot(v) # [7,5] (matrix-vector mult)
```

NOTEBOOK TIME

I'll build a Python notebook demonstrating some basic features of numpy.

After lecture it will be available here.

INDEXING AND SLICING

Numpy has powerful syntax for retrieving individual elements or collections of elements of arrays.

Most basic version: `A[i,j]` gives the element at row `i`, column `j` for a 2D array. Similar in higher dimensions, e.g. `A[i,j,k,l]`.

Slices return *views* of part of the array, not copies.

UFUNCS

Numpy's "ufuncs" or **universal functions** are functions that can be applied directly to arrays, automatically acting on each element.

Numpy provides a *lot* of these.

Usually, ufuncs allow you to avoid explicit iteration over array elements (which is much slower).

REFERENCES

- *Python Data Science Handbook* by Jake VanderPlas
    - Bookmark it now! We'll use it for several topics.
    - Chapter 2 contains the introduction to numpy.
    - There is also a print edition from O'Reilly.

REVISION HISTORY

- 2021-03-11 Move unused slides to Lecture 26
- 2021-03-10 Typo
- 2021-03-10 Initial publication