NUMPY

CS 010

Design and Implementation of Solutions to Computational Problems

Prof. Donald J. Patterson

EXTENDING PYTHON

MODULES EXTEND PYTHON

- Examples
  - ezgraphics
    - enabled us to draw pictures
  - re
    - regular expression module
- our own imported code
  - ComboLock

EXTENDING PYTHON

VERY WIDELY USED PYTHON EXTENSIONS

- Numpy
  - An n-dimensional matrix library
- Scipy
  - scientific computing library
- Matplotlib
  - 2D plots and graphs

NUMPY

THE BASICS

- The main thing that Numpy does is handle n-dimensional matrices
  - homogenous
    - all the elements are the same type
- We saw python to this natively with “tables”
  - lists of lists
- Numpy does it more efficiently and with an easier programming interface

NUMPY - INSTALLING

- Numpy is not installed when you install python
  - It is a module or “extension”

- Python comes with several different extension management systems
  - These simplify the process of adding extensions to your hard drive

NUMPY - INSTALLING

- You can download an extension as a ".py" file, but your python programs might not be able to find them
- a package manager puts the file in the right place so that it can be found
- pip3 is a python manager for python3

```
$ pip3 install numpy
Collecting numpy
  Downloading numpy-1.11.2-cp35-cp35m-macosx_10_6_intel.macosx_10_9_intel.macosx_10_9_x86_64.ma
cosx_10_10_intel.macosx_10_10_x86_64.whl (3.9MB)
    100% |████████████████████████████████| 3.9MB 156kB/s
Installing collected packages: numpy
Successfully installed numpy-1.11.2
You are using pip version 8.1.2, however version 9.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

PIP3

```
$ pip3 --help

Usage:
  pip <command> [options]

Commands:
  install                     Install packages.
  download                    Download packages.
  uninstall                   Uninstall packages.
  freeze                      Output installed packages in requirements format.
  list                        List installed packages.
  show                        Show information about installed packages.
  check                       Verify installed packages have compatible dependencies.
  search                      Search PyPI for packages.
  wheel                       Build wheels from your requirements.
  hash                        Compute hashes of package archives.
  completion                  A helper command used for command completion.
  help                        Show help for commands.

General Options:
  -h, --help                  Show help.
  --isolated                  Run pip in an isolated mode, ignoring environment variables and
                              user configuration.
  -v, --verbose               Give more output. Option is additive, and can be used up to 3
                              times.
  -V, --version               Show version and exit.
  -q, --quiet                 Give less output. Option is additive, and can be used up to 3
                              times (corresponding to WARNING, ERROR, and CRITICAL logging
                              levels).
  --log <path>                Path to a verbose appending log.
  --proxy <proxy>             Specify a proxy in the form [user:passwd@]proxy.server:port.
  --retries <retries>         Maximum number of retries each connection should attempt
                              (default 5 times).
  --timeout <sec>             Set the socket timeout (default 15 seconds).
  --exists-action <action>    Default action when a path already exists: (s)witch, (i)gnore,
                              (w)ipe, (b)ackup, (a)bort.
  --trusted-host <hostname>   Mark this host as trusted, even though it does not have valid
                              or any HTTPS.
  --cert <path>               Path to alternate CA bundle.
  --client-cert <path>        Path to SSL client certificate, a single file containing the
                              private key and the certificate in PEM format.
  --cache-dir <dir>           Store the cache data in <dir>.
  --no-cache-dir              Disable the cache.
  --disable-pip-version-check
                              Don't periodically check PyPI to determine whether a new
                              version of pip is available for download. Implied with --no-
```

WHERE TO FIND MORE ABOUT THIS

https://docs.scipy.org/doc/numpy-dev/user/quickstart.html

NUMPY - THE BASICS

NDARRAY

- ndarray.ndim
  - the number of axes (dimensions) of the array.
- ndarray.shape
  - the dimensions of the array. This is a tuple of integers indicating the size of the array in each dimension. For a matrix with n rows and m columns, shape will be (n,m). The length of the shape tuple is therefore the rank, or number of dimensions, ndim.

see scipy.org

NUMPY - THE BASICS

NDARRAY

- ndarray.size
  - the total number of elements of the array. This is equal to the product of the elements of shape.
- ndarray.dtype
  - an object describing the type of the elements in the array.
- ndarray.itemsize
  - the size in bytes of each element of the array.

see scipy.org

NUMPY - THE BASICS

NDARRAY

- ndarray.data
  - the buffer containing the actual elements of the array. Normally, we won't need to use this attribute because we will access the elements in an array using indexing facilities.

see scipy.org

NUMPY - THE BASICS

EXAMPLES

```
>>> import numpy as np

>>> c = np.array([[1,2,3,4,5],[6,7,8,9,10]])
>>> c.size
10
>>> c.shape
(2, 5)
>>> c = np.array([[1,2,3,4,5],[6,7,8,9,10]])
>>> c.ndim
2
>>> c.shape
(2, 5)
>>> c.size
10
>>> c.dtype
dtype('int64')
>>> c.dtype.name
'int64'
>>> c.itemsize
8
>>> c.data
<memory at 0x104030048>
>>>
```

see scipy.org

NUMPY - ARRAY CREATION

- ndarray.array
  - The easiest way is to leverage standard python
  - The type of the resulting array is deduced from the type of the elements in the sequences.

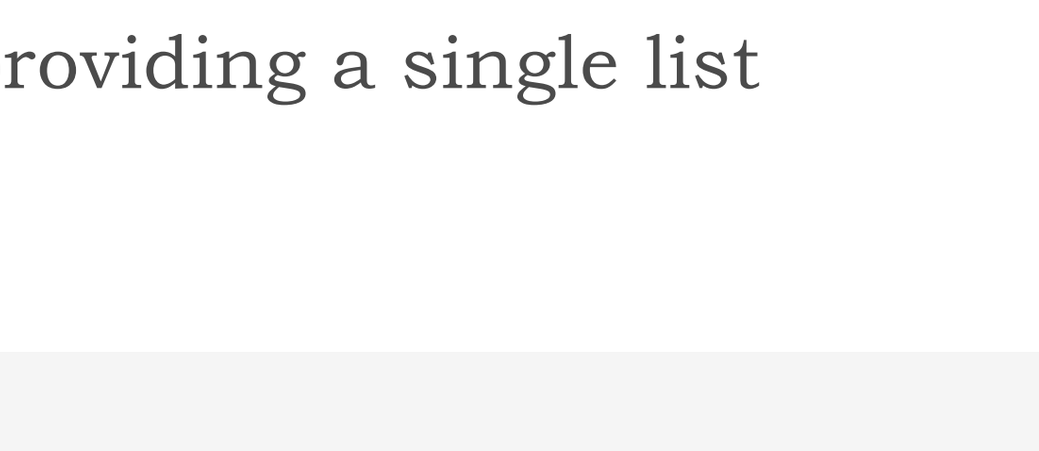
```
>>> import numpy as np
>>> a = np.array([2,3,4])
>>> a
array([2, 3, 4])
>>> a.dtype
dtype('int64')
>>> b = np.array([1.2, 3.5, 5.1])
>>> b.dtype
dtype('float64')
```

see scipy.org

NUMPY - ARRAY CREATION

- A frequent error consists in calling array with multiple numeric arguments, rather than providing a single list of numbers as an argument.

```
>>> a = np.array(1,2,3,4)     # WRONG
>>> a = np.array([1,2,3,4])   # RIGHT
```



see scipy.org

NUMPY - ARRAY CREATION

- array transforms sequences of sequences into two-dimensional arrays, sequences of sequences of sequences into three-dimensional arrays, and so on.

```
>>> b = np.array([(1.5,2,3), (4,5,6)])
>>> b
array([[ 1.5,  2. ,  3. ],
       [ 4. ,  5. ,  6. ]])
```


see scipy.org

NUMPY - ARRAY CREATION

- The type of the array can also be explicitly specified at creation time:

```
>>> c = np.array( [ [1,2], [3,4] ], dtype=complex )
>>> c
array([[ 1.+0.j,  2.+0.j],
       [ 3.+0.j,  4.+0.j]])
```


see scipy.org

NUMPY - ARRAY CREATION

- Often, the elements of an array are originally unknown, but its size is known. Hence, NumPy offers several functions to create arrays with initial placeholder content. These minimize the necessity of growing arrays, an expensive operation.
- The function zeros creates an array full of zeros, the function ones creates an array full of ones, and the function empty creates an array whose initial content is random and depends on the state of the memory. By default, the dtype of the created array is float64.

see scipy.org

NUMPY - ARRAY CREATION

```
                                                                    >>>
>>> np.zeros( (3,4) )
array([[ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.]])
>>> np.ones( (2,3,4), dtype=np.int16 )                # dtype can also be spe
cified
array([[[ 1, 1, 1, 1],
        [ 1, 1, 1, 1],
        [ 1, 1, 1, 1]],
       [[ 1, 1, 1, 1],
        [ 1, 1, 1, 1],
        [ 1, 1, 1, 1]]], dtype=int16)
>>> np.empty( (2,3) )                                 # uninitialized, output
 may vary
array([[  3.73603959e-262,   6.02658058e-154,   6.55490914e-260],
       [  5.30498948e-313,   3.14673309e-307,   1.00000000e+000]])
```

see scipy.org

NUMPY - ARRAY CREATION

- To create sequences of numbers, NumPy provides a function analogous to range that returns arrays instead of lists

```
>>> np.arange( 10, 30, 5 )
array([10, 15, 20, 25])
>>> np.arange( 0, 2, 0.3 )                 # it accepts float arguments
array([ 0. ,  0.3,  0.6,  0.9,  1.2,  1.5,  1.8])
```

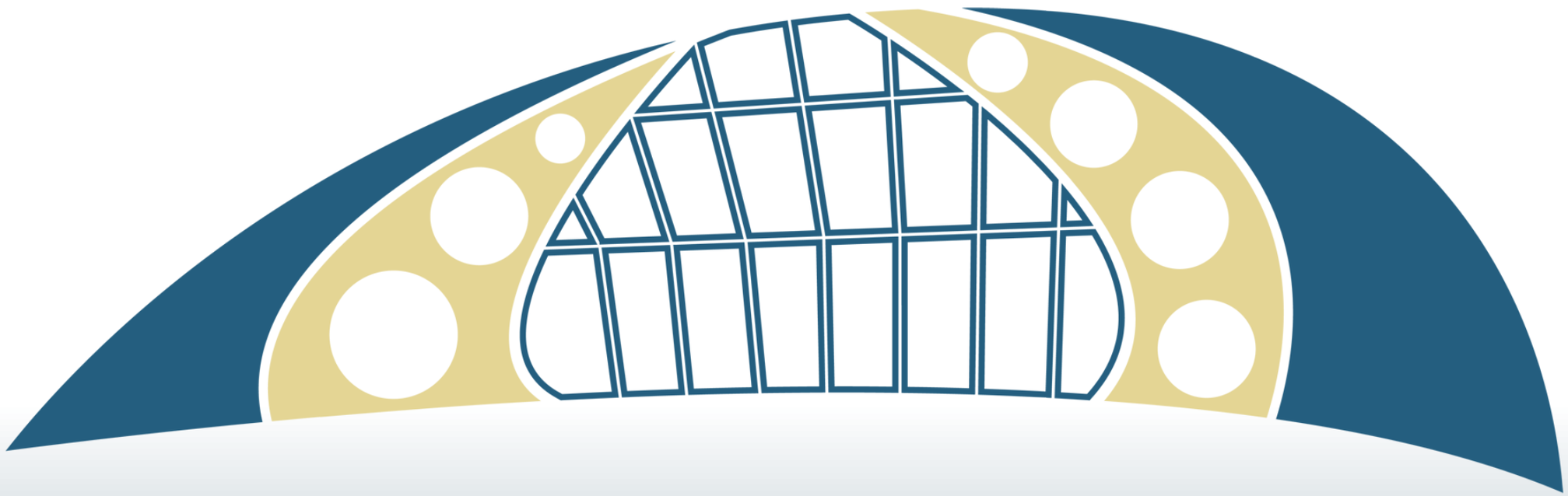

see scipy.org

NUMPY - ARRAY CREATION

- When arange is used with floating point arguments, it is generally not possible to predict the number of elements obtained, due to the finite floating point precision. For this reason, it is usually better to use the function linspace that receives as an argument the number of elements that we want, instead of the step:

```
>>> from numpy import pi
>>> np.linspace( 0, 2, 9 )                 # 9 numbers from 0 to 2
array([ 0.  ,  0.25,  0.5 ,  0.75,  1.  ,  1.25,  1.5 ,  1.75,  2.  ])
>>> x = np.linspace( 0, 2*pi, 100 )        # useful to evaluate function at l
ots of points
>>> f = np.sin(x)
```

see scipy.org



WESTMONT INSPIRED

COMPUTING LAB