Packages and modules

Terence Parr
MSDS program
University of San Francisco

Why we need the import statement

- We've seen the use of some predefined functions, such as **range()** and **len ()** that are available without doing anything special in your Python program
- Now let's take a look at importing and using code libraries
- There are a multitude of of libraries on your disk and Python can't automatically load them all into memory; we must explicitly **import** the libraries we want to use in our program
- This is like opening a specific cookbook of recipes
- Later we will see how to install more libraries onto our machine

Accessing goodies from installed modules

- Imagine that we need the constant π; if we try referencing **pi**, we find that it's not defined, but of course we could define one:

```
>>> pi
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'pi' is not defined
>>> pi = 3.14
>>> pi
3.14
```

- Easier to access a pre-installed and available library:

```
>>> import math
>>> math.pi
3.141592653589793
>>>
```

Somewhere there's a **math.py** *file*

***Warning**: '.' is overloaded so a.b could mean package, module, or object accessor*

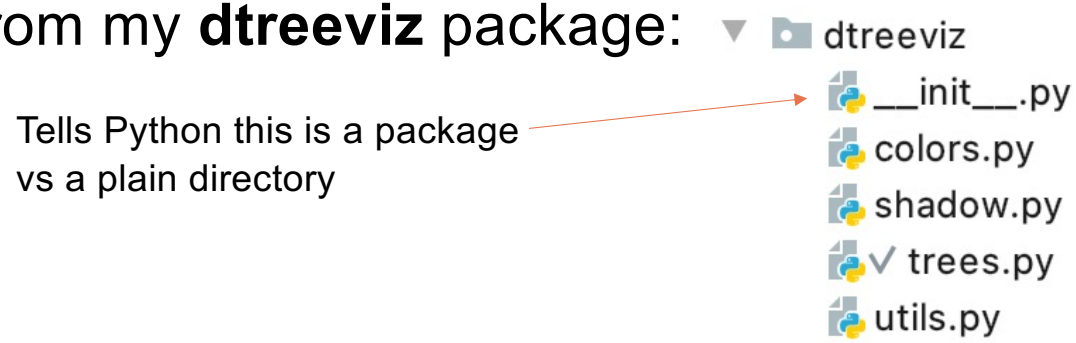
What's in a module?

- Use **dir()** to look at module contents (or google)
- There are functions as well as variables in **math.py**:

```
>>> import math
>>> dir(math)
['__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'ac
os', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'comb', 'copysi
gn', 'cos', 'cosh', 'degrees', 'dist', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs
', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf'
, 'isclose', 'isfinite', 'isinf', 'isnan', 'isqrt', 'ldexp', 'lgamma', 'log', 'l
og10', 'log1p', 'log2', 'modf', 'nan', 'perm', 'pi', 'pow', 'prod', 'radians', '
remainder', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']
>>> math.cos(math.pi)
-1.0
```

Some key terminology

- A script .py file is also called a *module*; if **a.py** imports module **b** then **a.py** can access the variables and functions in file **b.py**
- A directory containing module file(s) is called a *package*; if directory **p** contains module file **m.py**, then a script like **foo.py** can **import p.m** to access the goodies from **m.py**
- Example from my **dtreeviz** package:



Example: **numpy** package

- NumPy is a library whose outermost package is called **numpy** with a **random** module that has a **randint()** function:

```
>>> import numpy
>>> numpy
<module 'numpy' from '/Users/parrt/opt/anaconda3/lib/python3.8/site
-packages/numpy/__init__.py'>
>>> numpy.random.randint
<built-in method randint of numpy.random.mtrand.RandomState object
at 0x7f9df83ce240>
>>> numpy.random.randint(100,110)
104
```

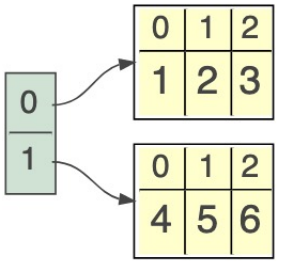
Installing more libraries

- We recommend that you install Anaconda's Python bundle, which includes most of the stuff we need for machine learning
- We can **import** any of the predefined Python libraries or the preinstalled libraries in Anaconda
- There are a huge number of useful Python packages that are likely not currently installed on your machine; this is analogous to all of the uninstalled apps you see in the app store
- To install library foo: **pip install foo**
- There is another package manager called **conda**, which is more sophisticated than pip (can installed C++ code, etc…)

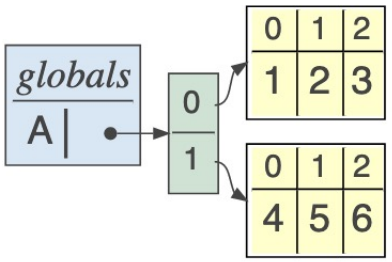
Example: installing lolviz

- lolviz is a package I built to display data structures
- First, use the terminal to install **graphviz**, a program needed by my lolviz library: **brew install graphviz**
- Then, **pip install lolviz**

```
import lolviz
A = [[1,2,3],[4,5,6]]
lolviz.objviz(A)
```




```
lolviz.callsviz(varnames=['A'])
```



Fancier notation

- With some new notation, we can create a shorthand to access functions of interest more easily

  ```
  from module import x, y, …, z
  ```

- It's also very handy to create a shorter name for a package:

  ```
  import numpy as np
  import pandas as pd
  A = np.sum([1,2,3])
  ```

```
from lolviz import objviz
A = [[1,2,3],[4,5,6]]
objviz(A)
```