Modules and Files

Chapter 7

Modules

- Allows a group a definitions related to the same topic to be grouped
- Stored in a file together
  - Like a “library”
  - Often referred to as a package
- Included in a program with the **`import`** keyword
- Referenced using “dot” notation
  - `module.definition`

Modules
Alternate form

- Alternate form
  - `from` *`module`* `import *`
  - Dot notations is not required
  - Can cause naming conflicts

Module alias

- Can create an “easy to use” name
  - Uses the **`as`** keyword

  - `import really_long_module_name as rlm`

Module example

- Filename could be “circle.py”
- `import circle`
- `circle.pi`

```
pi = 3.14159

def area(radius):
    return pi*(radius**2)

def circumference(radius):
    return 2*pi*radius

def sphereSurface(radius):
    return 4.0*area(radius)

def sphereVolume(radius):
    return (4.0/3.0)*pi*(radius**3)
```

Predefined modules

- Python standard library contains hundreds of modules
  - We have already seen `copy`
  - We will explore several but not nearly all of them
  - Other examples; `math` and `cal`
- Python Package Index (PyPI)
  - Contains hundreds of *thousands* of modules
  - Need to download and install
  - https://pypi.org/

Import conventions

- Use import not with
- One import per line
- Place all imports at start of program
- Start with system defined modules
- Follow with 3rd party modules (Anaconda or PyPI)
- Finally, include local, or project, modules

Files

- **open(fn, 'w')** fn is a string representing a file name. Creates a file for writing and returns a file handle.
- **open(fn, 'r')** fn is a string representing a file name. Opens an existing file for reading and returns a file handle.
- **open(fn, 'a')** fn is a string representing a file name. Opens an existing file for appending and returns a file handle.
- **fh.read()** returns a string containing the contents of the file associated with the file handle fh.
- **fh.readline()** returns the next line in the file associated with the file handle fh.
- **fh.readlines()** returns a list each element of which is one line of the file associated with the file handle fh.
- **fh.write(s)** write the string s to the end of the file associated with the file handle fh.
- **fh.writeLines(S)** S is a sequence of strings. Writes each element of S to the file associated with the file handle fh.
- **fh.close()** closes the file associated with the file handle fh.

More on Files

- We’ll often work with files
- More on files in coming weeks