Scripting Programming Languages and their Applications

Jan Gaura

October 2, 2012

Introspection

- introspection (reflection) is super powerful
- let's check one useful trick
- used in modules testing

```
1   def main():
        '''just func def'''
3       pass

5   if __name__ == "__main__":
        main()
```

Importing modules

- `import modulename`
- `from some.given.package import moduleattrib`
  - then, in either case, use modulename.blah
  - some abbreviations (not necessarily recommended...):
    - alias module name with an 'as' clause:
    - `import thisnameisfartoolong as z`
    - then use z.blah
    - `from thisnameisfartoolong import blah`
    - `from thisnameisfartoolong import *` # **not** recommended
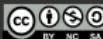
Import Example

```
   import math
 2 print math.atan2(1, 3)
   # emits 0.321750554397

 4

   print atan2(1, 3)
 6 # raises a NameError exception

 8 from math import atan2
   # injects atan2 into current namespace
10 # handy for interactive use, often
   # confusing in "real" programs
12 # and, always avoid (except for interactive use)...:

14 from math import *
```
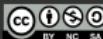
Coding Your Own Modules

- any Python source `wot.py` **is** a module
- just `import wot`
    - file must be in a directory (or zip file) listed in `sys.path`
    - also importable: the bytecode file (wot.pyc) which the Python compiler creates the first time you import the source
    - also importable: binary native-code extension files (wot.pyd) coded in C and using Python's C API (or other compiled-to-native-code languages and tools: pyrex, Cython, SWIG, ...)
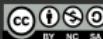
Packages

- a package is a module containing other modules (and maybe sub-packages &c)
- lives in a directory containing a file named `__init__.py`
    - `__init__.py` is the "module body"
    - often empty (just "tagging" the directory as a package)
    - a package's module are the dir's `.py` files
    - subpackages are subdirs w/`__init__.py`
- the parent-directory must be on sys.path
- import foo.bar or from foo import bar

References

- Alex Martelli, Painless Python for Proficient Programmers
- Django documentation
- Adam Fast, intro to geodjango