IMPORTING FUNCTIONS

Python Programming — Auburn University

MODULES

- Python code, even in the standard library, is broken up into *modules*.

- A python module is similar to a java package (java.util).

  - Be careful, Python also has "package" concept which we'll discuss more later.

- Modules provide a "namespacing" mechanism to prevent name conflicts from separate libraries.

  - For example, you're free to create a class called `Component` even though there may be many classes with the same name in other modules.

IMPORT

- To reference values (functions, classes, constants etc) defined in a module, you must import the module using an import statement:

  - `import math`

- Once imported, you can reference values in the module using `module_name.symbol`:

  - `math.gcd(34345897384, 12)`

THE FROM-IMPORT STATEMENT

- If you prefer to refer to a value without its module name, and assuming you have no conflicting names in your module or imported into your module, you can use from-import:

    - ```
      from math import gcd
      gcd(34345897384, 12)
      ```

- Use from-import-as if you prefer to use a different name (or to avoid conflict) to reference the value:

    - ```
      from math import gcd as monkey
      monkey(34345897384, 12)
      ```

MODULE INFORMATION IN DOCUMENTATION

- Demo: Documentation of `datetime`

USING "AS"

- Using the "as" keyword allows you to reference an object by a different name:

```
from math import sqrt as square_root
print(square_root(3))
```

- Used to avoid conflicts (two packages with same function name) or to make names more readable in the context in which they are being used.

MORE ON MODULES/PACKAGES LATER…

- Where are modules stored?

  - A Python module is typically stored in a single file named with ".py" added to the module name.

  - There is a **configurable** search path that Python uses to find modules.

- How do I install 3rd party modules?

  - The python package manager "pip" is the most common way.

- How do I create my own module/package? We'll cover this soon!