

INFO5002: Intro to Python for Info Sys

Modules & Packages

PCC 149-152, 173-179



**Northeastern
University**

Terminology

- A **module** is a Python file w/ reusable elements.
 - Classes and functions
- A **package** is a folder with at least 1 module and a special `__init__.py` file.
- You can nest packages into packages which become **sub-packages**.

Example

/		
algorithms/	←	code directory
__init__.py	←	package
euclidean.py	←	special init file
math/	←	module
__init__.py	←	sub-package
basic.py	←	special init file
main.py	←	module
screen.py	←	entry point
	←	not a module

Use init to expose

```
/
  algorithms/
    __init__.py
    euclidean.py
    math/
      __init__.py
      basic.py
  main.py
  screen.py
```

```
# /algorithms/math/basic.py
import math

def add(x, y):
    """Add y to x"""
    return x + y
def sub(x, y):
    """Subtract y from x"""
    return x - y
def mul(x, y):
    """Multiply x by y"""
    return x * y
def div(x, y):
    """Divide x by y"""
    return x / y
def sqrt(x):
    """Square root x"""
    return math.sqrt(x)
```

```
# /algorithms/math/__init__.py

from .basic import add, sub,
mul, div, sqrt
```

```
# /algorithms/euclidean.py

from .math import mul, sqrt,
add

def hypotenuse(x, y):
    return sqrt(add(mul(x, x),
                    mul(y, y)))
```

Import package as normal

```
/
algorithms/
  __init__.py
  euclidean.py
  math/
    __init__.py
    basic.py
main.py
screen.py
```

```
# /main.py

from algorithms import
hypotenuse, add

print("The hypotenuse of
the 3, 4 triangle is: " +
str(hypotenuse(3, 4)))

print("1 + 1 = " +
str(add(1, 1)))
```

```
# /algorithms/__init__.py
```

```
from .euclidean import
hypotenuse
from .math import add
```

```
# /algorithms/euclidean.py
```

```
from .math import mul, sqrt,
add
```

```
def hypotenuse(x, y):
    return sqrt(add(mul(x, x),
mul(y, y)))
```

We import with from import

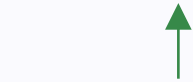
from keyword



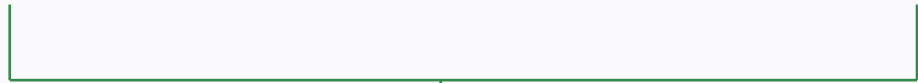
import keyword



```
from package import module1, module2, module3, ...
```



package name



package's module names

Different packages

- Import **python** std packages via their name

```
from math
```

- Import **external** packages via their name

```
from numpy
```

- Will need to install first with pip

```
pip install numpy
```

- Can see all packages from pypi.org

```
from algorithms.math
```

- Import user defined packages via absolute

```
from .math
```

- Import user defined packages via relative

I want to import everything

```
from package import *
```



* wildcard

Entry file special case

- The entry file is not loaded as a module and for that reason when you run it you cannot use relative imports as it **is** the start.
- Always use absolute paths.

Let's practice

calculator_project/

├─ main.py

└─ calculator/

 ├─ __init__.py

 ├─ add.py

 ├─ subtract.py

 ├─ multiply.py

 └─ divide.py

Design the following project on your computer where when **main** executes it asks the user to input a first number, second number, and finally the operation to perform.