University of
South Australia

Problem Solving and Programming

More on Modules
Reading Slides

Python Books

Course Textbook

Gaddis, Tony. 2012, *Starting Out with Python*, 2nd edition, Pearson Education, Inc.

Free Electronic Books

There are a number of good free on-line Python books. I recommend that you look at most and see if there is one that you enjoy reading. I find that some books just put me to sleep, while others I enjoy reading. You may enjoy quite a different style of book to me, so just because I say I like a book does not mean it is the one that is best for you to read.

- The following three books start from scratch - they don't assume you have done any prior programming.
    - The free on-line book "**How to think like a Computer Scientist: Learning with Python**" 2nd edition, by Allen B. Downey and Chris Meyers, provides a good introduction to programming and the Python language. I recommend that you look at this book.
    - There is an on-line book "**A Byte of Python**" that is quite reasonable. See the home page for the book, or you can go directly to the on-line version for Python 3, or download a PDF copy of the book. This book is used in a number of Python courses at different universities and is another I recommend you look at.
    - Another good on-line book is "**Learning to Program**" by Alan Gauld. You can download the whole book in easy to print PDF format, and this is another book that would be good for you to look at.
- If you have done some programming before, you may like to look at the following:
    - **The Python Tutorial** - this is part of Python's documentation and is updated with each release of Python. This is not strictly an eBook, but is book-sized.
    - **Dive into Python 3**, by Mark Pilgrim is a good book for those with some programming experience. I recommend you have a look at it. You can download a PDF copy.

Standard Library Functions

- Python has an extensive library of functions.
  - Built-in functions
    - The Python interpreter has a number of functions that are always available.
    - They are available without having to import a library.
  - Many of the functions in the standard library are stored in files called modules.
  - Modules organise the standard library functions.
    - For example:
      - Functions for performing math operations are stored together in the `math` module.
      - Functions for generating random numbers are stored together in the `random` module.
  - To call a function that is stored in a module, you have to write an import statement at the top of your program.
  - An import statement tells the interpreter the name of the module that contains the functions.

Standard Library Functions

`math` – Mathematical functions

- Provide access to the mathematical functions.
- Need to place the following import statement at the top of your program.

```
import math
```

- The import statement causes the interpreter to load the contents of the `math` module into memory and makes the functions in the `math` module available to the program.

Standard Library Functions

`random` – generate pseudo-random numbers

- It provides access to random number generator.
- Need to place the following import statement at the top of your program.

```
import random
```

- The import statement causes the interpreter to load the contents of the `random` module into memory and makes the functions in the `random` module available to the program.

Standard Library Functions

- A module is a file that contains Python code.
- As programs become larger and more complex, the need to organize code becomes greater.
  - Related functions may be organised by storing them in modules.
  - Each module should contain functions that perform related tasks.
  - This approach is called modularization.
- Modules make a program easier to understand, test and maintain.
- Modules also make it easier to reuse the same code in more than one program.
  - Place related functions that are needed in several programs in a module.
  - Import the module in each program that needs to call one of the functions.

Storing Functions in Modules

- An example:
  - Suppose we need to write functions that calculate the following:
    - The area of a circle
    - The circumference of a circle
  - We can place the circle related functions in a module called `circle.py` like so:

```
# The circle module has functions that perform
# calculations related to circles.
import math

# The area function accepts a circle's radius as an
# argument and returns the area of the circle.
def area(radius):
    return math.pi * radius**2

# The circumference function accepts a circle's
# radius and returns the circle's circumference.
def circumference(radius):
    return 2 * math.pi * radius
```

Tony Gaddis. 2012, *Starting Out with Python*, 2nd Edition, Pearson Education, Inc.

Storing Functions in Modules

- An example *(continued)*:
  - The `circle.py` file contains function definitions, but it does not contain code that calls the functions. That will be done by the program(s) that import the `circle` module.
  - Please note:
    - A module's file name should end in *.py. If the module's file name does not end in *.py you will not be able to import it into other programs.
    - A module's name cannot be the same as a Python keyword.
  - To use the modules in a program, import them with the import statement (just like the functions available in the Python Standard Library).
  - To import the `circle` module:

```
import circle
```

When the Python interpreter reads this statement, it will look for the file `circle.py` in the same folder as the program that is trying to import it. If found, it will load it into memory. If not found, an error occurs.

Storing Functions in Modules

- An example *(continued)*:
  - Once a module is imported, you can call its functions.
  - Here is an example of a program that uses the `circle.py` module:

```
# Import the circle module
import circle

radius = 10

my_area = circle.area(radius)
my_circ = circle.circumference(radius)

print('The area is:', my_area)
print('The circumference is:', my_circ)
```

Tony Gaddis. 2012, *Starting Out with Python*, 2nd Edition, Pearson Education, Inc.

End of Reading Slides