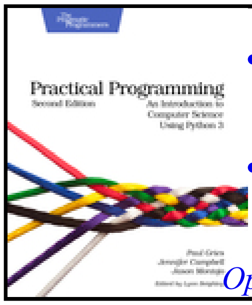
Import: Using Non-Builtin Functions



- 6.1 Importing Modules
- 6.2 Defining Your Own Modules

Optional reading

Modules

Python contains many functions, but not all of them are immediately available as builtin functions. Instead of being available as builtins, some functions are saved in different modules. A *module* is a file containing function definitions and other statements.

We may also define our own modules with our own functions.

`import`

In order to gain access to the functions in a module, we must import that module.

The general form of an import statement is:

```
import module_name
```

To access a function within a module, we use:

```
module_name.function_name
```

For example, we can import the Python module `math` and call the function `sqrt` from it:

```
import math

def area2(side1, side2, side3):
        semi = semiperimeter(side1, side2, side3)
        area = math.sqrt(semi * (semi - side1) * (semi - side2) * (semi - side3))
        return area
```

In addition to importing Python's modules, we can also import the modules that we write. For example, to use the functions from `triangle.py` (from the video) in another module, we would `import triangle`. A module being imported should be in the same directory as the module importing it.

Jennifer Campbell • Paul Gries
University of Toronto