Documentation and Elements of a Program

23 Sep 2010
CMPT140
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Binary arithmetic operators

- + , –, *: addition, subtraction, multiplication
- **: power: 2**4 → 16
- /: division: 7.0 / 2 → 3.5
  - In many languages, integer '/' is floor division!
- //: floor division (floor of quotient)
  - On ints: 7 // 2 → 3
  - On floats: 7.0 // 2 → 3.0
- %: modulo (remainder): 8 % 3 → 2
  - 8 % 0 → ZeroDivisionError

Comparison operators

- Test for quantitative equality: 2 + 3 == 5
- Test for inequality: 2 + 3 != 4
- Comparison: <, >, <= , >=
- Test for identity: is, is not
  - (2, 3) == ((2, 3)), but
  - (2, 3) is not ((2, 3))

Boolean operators: shortcut

- Boolean operators: and or not
  - In C/C++/Java: && || !
- Boolean operators have shortcut semantics:
  - Second operand is only evaluated if necessary
    - (7 / 0) and False → ZeroDivisionError
    - False and (7 / 0) → False
      - Doesn't raise ZeroDivisionError
    - True or (7 / 0) → True
      - Same thing

Precedence

- Order of evaluation from highest (evaluated first) to lowest (evaluated last) is
  - **
  - **Unary +, -**
  - *, /, %, //
  - **Binary +, -**
  - ==, !=, <>, <, >, <=, >=
  - **is, is not**
  - **not**
  - **and**
  - **or**
- Complete precedence rules at http://docs.python.org/ref/summary.html

Strings and quoting

- Strings in Python can be in either 'single' or “double” quotes
- What if you want a quote mark in your string?
  - “It is I; don't be afraid”
  - 'Jesus said, “I am the way, and the truth, and the life.” '
- To include a newline (carriage return) in string, use three double-quotes:
  - **”””This is a multi-line string.**
    Even the newline is part of the string.”””
  - This is rather special to Python!

Documentation

- Document your thinking at every step,
  even the ideas that didn't work!
  - Programmer's diary: log of everything
- External documentation: outside the program
  - User manual:
    - What user input is required
    - What the user should expect the program to output
    - No details about program internals
- Internal documentation: within the program
  - Descriptive variable/module names
  - Comments in the code
  - Online help for the user

Internal documentation

- Good variable names: numHashes
  - Bad variable names: x, num, i
- Comments: # in Python (to end of line)
  - **# loop numHashes times**
  - **while (counter < numHashes):**
    - **print "#", # no newline**
    - **counter = counter + 1**
- Online help:
  - "Enter 'h' for online help."

Comments

- Explain the "why", not the "what":
  - Bad: `x = x + 1  # increment x`
  - Good: `x = x + 1  # do next hashmark`
- Keep comments up-to-date!
  - Incorrect comments are worse than no comments
- Comments are no substitute for external documentation
  - Still need a separate design doc, pseudocode, user manual, etc.

Docstrings

- Python convention is to create a docstring at the top of every module, function, class, etc.:
  - **” ” ” Print a bunch of hashes.**

    Nellie Hacker, CMPT140
    ” ” ”
    numHashes = input(“How many hashes? ”)
    . . .
- Triple-quotes: this is a string, not a comment
- First line is a short summary
- Second line is blank, then detailed description
- Automated Python tools read docstrings to help you organize your code
  - More info: http://www.python.org/dev/peps/pep-0257/

Style conventions

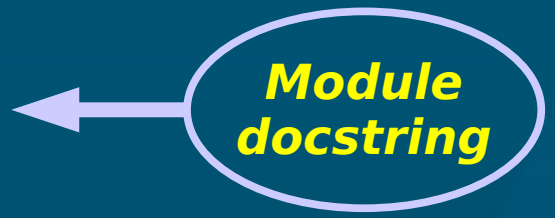
- Not hard-and-fast rules, but flexible conventions that make code easier to read and understand
- Variable names: numHashes
  - Flexible, but I prefer no underscores, and capitalize each word ("CamelCase")
  - First letter is lowercase
- File/module names: helloworld.py
  - Short, all lowercase, no underscores
- Function names: print_hashes()
  - lowercase, command predicate, underscores
- More details: http://www.python.org/dev/peps/pep-0008/

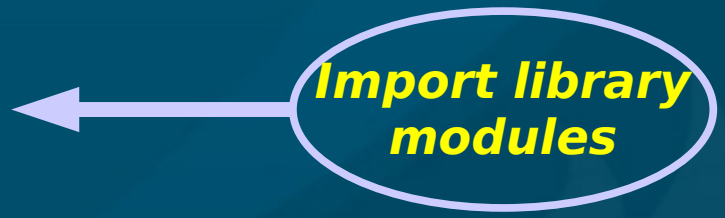
Components of “helloworld.py”

```
"""A baby Python program.

Name: John Doe
This is a sample program.
"""
import math

print "Hello World!"
print "Pi =", math.pi
```





Modules

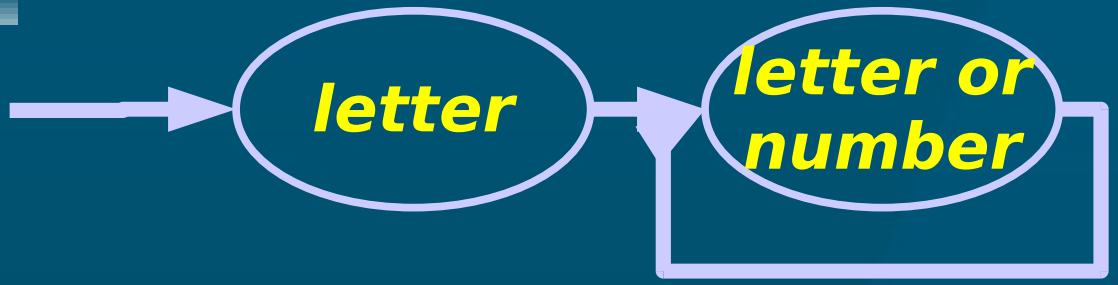
- A module is a container holding items (vars, functions, …) constituting all or part of an executable program
- In Python, every file is a module
  - helloworld.py is a module that is also an executable script
- math is a library module from which we imported the pi constant
- math.pi is not a module but a name within a module

Identifiers

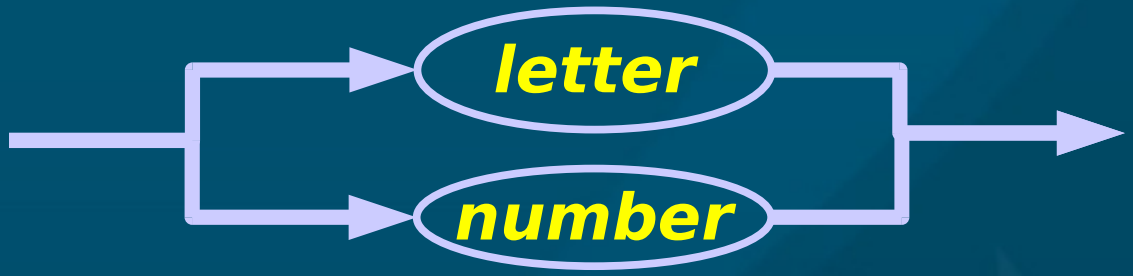
- Identifiers are names for stuff: e.g.,
  - Libraries (“math”), functions (“print”), variables (“numApples”)
- Python identifiers must be sequences of
  - Letters (case-sensitive) or digits
  - Must start with a letter
    - (underscore _ counts as a letter)
- OK: Great_Googly_Moogly, x, My21stBirthday
- Not OK: “hi ya”, h@Xz0r, 21stBirthday
- These are the rules; we'll talk about style later

Railroad diagram for identifiers

- identifier =



- letter or
  number =



- number = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
- letter = {a, b, ..., z, A, B, ..., Z, _}

Literals vs. identifiers

- A literal is an entity whose name is an encoding of its value:
  - 187.3
  - “Hello World!”
  - True
- In contrast, the value of a variable may change even though its name stays the same:
  - numApples = 5
  - numApples = 6

Reserved words

- You can name your modules, functions, and variables almost anything you want, except
- Reserved words (keywords): special words or markers used to outline the structure of a program
  - import, if, else, while, for, def, ...
  - Complete list at
    http://docs.python.org/ref/keywords.html

Importing library functions

- Library functions are building blocks:
  - Tools that others wrote that you can use
- Functions are grouped into libraries:
  - If you want to use a pre-written function, you need to specify which library to import it from

```
import math
math.sqrt( 2 )          >>>  1.4142135623730951
math.pow( 3, 5 )        >>>  243.0
math.pi                 >>>  3.1415926535897931
```

Python Standard Library

- Library functions provided with every standard Python implementation
- You still have to import them, though
- Our HelloWorld.py program used pi from the math standard library

- There are oodles of standard library functions:
  http://docs.python.org/lib/lib.html