Objectives

- Review: Linux, documentation, programming
- More on programming in Python
- Software development practices
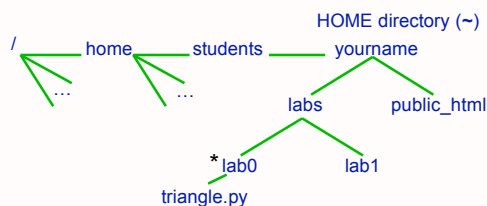 - Testing
  - Debugging
  - Iteration

Review: Linux

- How do you …
  - Display the files in a directory?
  - Go into a directory?
  - Create a directory?
  - Find out the current directory?
- What is the syntax for the copy command?
- What is the "abbreviation" for …
  - The current directory?
  - The parent directory?

Review: Linux File System

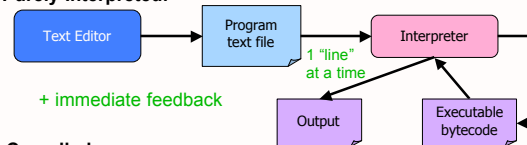~ is a shortname for your home directory, i.e., short for /home/students/yourname

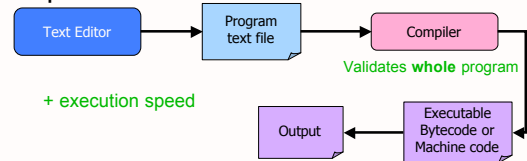How would you copy triangle.py to your public_html directory?

Compiled vs. Interpreted Languages

Purely Interpreted:

+ immediate feedback

Compiled:

Validates **whole** program

+ execution speed

Python's Implementation

- Combination of compiled and interpreted
- Interactive mode: interpreted
  - Validate, execute each line
- Python "interpreter" in script mode:
  - Compiles Python script into **bytecode**
  - Runs Python Virtual Machine that interprets the bytecode and executes

Documenting Your Code

- Use English to describe what your program is doing in **comments**
  - Everything after a **#** is a comment
    - Color-coded in IDLE, jEdit
  - Python does not execute comments
- Does not affect the correctness of your program
- Improves program's readability
  - Easier for someone else to read and update your code

1

When to Use Comments

- Document the author, high-level description of the program at the top of the program
- Provide an outline of an algorithm
  - Separates the steps of the algorithm
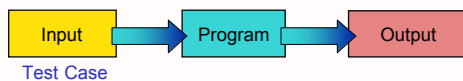- Describe difficult-to-understand code

Errors

- Sometimes the program doesn't work
- Types of programming errors:
  - Syntax error
    - Interpreter shows where the problem is
  - Logic/semantic error
    - answer = 2+3;
    - "The anser is "
  - Exceptions/Runtime errors
    - answer = 2/0;
    - Undefined variable name
- Expose when **Testing**
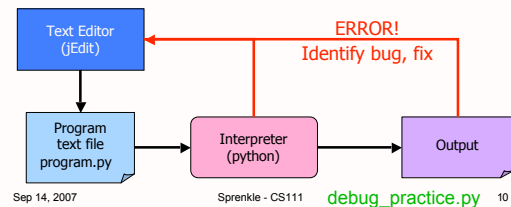
Testing Process

Input → Program → Output

Test Case

- Test case: input used to test the program
- Verify if output is what you **expected**
- Need *good* **test cases** to help determine if program is correct
  - Tester plays devil's advocate
  - Want to expose **all** bugs!
  - Find before customer/professor!

Debugging

- Edit the program, re-execute/test until everything works
- The error is often called a "bug"
- Diagnosing and fixing it is called **debugging**

Text Editor (jEdit) → Program text file program.py → Interpreter (python) → Output

ERROR! Identify bug, fix

Finishing Up Wednesday's Practice

- Average three numbers

Good Development Practices

- Design the algorithm
  - Break into pieces
- Implement and Test each piece separately
  - Identify the best pieces to make progress
  - Iterate over each step to improve it
- Write comments FIRST for each step
  - Elaborate on what you're doing in comments when necessary

Review: Numeric Primitive Types

Data Type	Description	Examples
int	Plain integers (32-bit precision)	-214, -2, 0, 2, 100 Range: -2^{31} to $2^{31}-1$
float	Real numbers	.001, -1.234, 1000.1, 0.00, 2.45
long	Bigger integers (neg or pos, precision limited by computer memory)	2147483648L
complex	Imaginary numbers (have real and imaginary part)	1j * 1J --> (-1+0j)

Strings: **str**

- Used for text
- Indicated by double quotes "" or single quotes ''
  - In general, I'll use double quotes
  - Empty string: "" or ''
- Use triple quotes """ for strings that go across multiple lines
  - """This string
    is long.
    Like, really long """

Printing Output

- **print**
  - Simple statement
    - _not_ a function
  - Displays the result of expression(s) to the terminal
- print "Hello, class"

  print automatically adds a '\n' (carriage return) after it's printed

  string literal　　*Unless ends with ,*
- print "Your answer is", answer

  Use commas to print multiple "things" in one line

Escape Sequences

- Escape Sequences
  - newline character (carriage return) -> \n
  - tab -> \t
  - quote -> \"
  - backslash -> \\
- Example:
  - print "To print a \\, you must use \"\\\\\""

Practice

- Print To print a tab, you must use '\t'.
- Print I said, "How are you?"

Getting Input From User

- **input** and **raw_input** are functions
  - Prompts user for input, gets the user's input
  - **input**: for numbers
  - **raw_input**: for strings
- Typically used in assignments　　*Prompt displayed to user*
  - width = input("Enter the width: ")
- In execution, terminal displays
  - "Enter the width: "
  - Assigns width the value the user enters

Getting Input from User

- color = raw_input("What is your favorite color? ")

Terminal:

Grabs every character up to the user presses "enter"

```
> python input_demo.py
What is your favorite color? blue
Cool!  My favorite color is _light_ blue !
```

Assigns variable **color** the user's input

input_demo.py

Sep 14, 2007   Sprenkle - CS111

Broader CS Issues

- Good summaries!
  - Good English, complete sentences

- Mechanics details
  - Post as **comments** to keep the blog a bit more organized
  - Follow instructions on "CS Issues" about what summary should contain
  - Can edit your own posts
  - May want to write in a word processor and then copy over to blog

Sep 14, 2007   Sprenkle - CS111

Four Problems of Cyberspace

- How many of you knew about MMOGs before reading this article?
  - How may of you participated in this or something similar?
- How many of you read online?
  - Printed out my PDF version?
  - Printed out the Web page?

Sep 14, 2007   Sprenkle - CS111

Four Problems of Cyberspace

- What is the main points (so far)?

- Every advancement in technology has positive and negative impacts
  - What are the positive and negative impacts of email? Of IM?

- Maya
- Oliver
- Alysen
- Will L.

- Jennifer
- Keith
- Cathy

- Laura
- Will R.
- Matt

Sep 14, 2007   Sprenkle - CS111