Outline

- Using IDLE
- Building blocks of programs:
  - Text
  - Numbers
  - Variables!
- Writing a program
- Running the program
Starting up Python

- From START menu, pick "all programs", then Python
- Pick the "IDLE" option
IDLE

- IDLE is an interpreter
- Can use it like a calculator
- Responds to input line-by-line
Remainder

- 0, 1, 2, 3... and -1, -2, -3... are integers
- 7//3 is integer division
- 7/3 is floating point division
- 7%2 = ?
- % gives the remainder when 7 is divided by 2
- (7//2)*2 + (7%2) = 7
Floating point numbers

- 7.0, 2.0, 0.0006, 7.34 – **floating point numbers**
- 7.0/2.0 = 3.5 – **floating point division**
- 7/2 = 3.5
- If either number is floating point, so is the answer – so 7.0/2 = 3.5
- 8.0/3.0 = 2.666...665?
- Floating point arithmetic does NOT give exact results!
Why not?

- Computer numbers have a fixed number of decimal places.
- Exact results with floating point numbers have an infinite number of decimal places:
  Example: \( 8.0/3.0 = 2.666666...... \)
Variables

- \( x = 2.0 \) -- \( x \) is a **variable**
- This is called an **assignment**
- Variable on left-hand side gets value on right-hand side.
- Pronounce this “\( x \) gets 2.0” or “\( x \) becomes 2.0”
- \( x = x+3.0 \) – “\( x \) gets \( x+3 \)” , so now \( x=5.0 \)
Variable Names

- Legitimate names:
  - Letter (upper & lower case), number, underscore
  - Do not start with a number
  - Ex: x, y, m1, m2, xysgfh, my_ID, my_age, myAge, etc.
  - Python is case-sensitive!
  - Python command in lower case.

- Good names
  - Easy to remember and to understand the meaning
  - Not too long, not too short
  - Ex: my_age, myAge, etc.
  - Be consistent, e.g., underscore, capitalize
Errors

- Lots of things you do will cause errors
- Something Python doesn’t understand
- $y = y+3$ – you ask it to give the value $y+3$ to $y$, but it doesn’t know what $y$ is.
- Variables don’t stand for “any old number” like they do in algebra; a variable is always supposed to have a specific value.
Python commands in IDLE

- You can type any Python command into IDLE, and it does it immediately
- In lower case.

- ‘a rose is a rose’ is a **string**
- “9.0/7.0” is also a string, because it’s in quotes
Making a program

- Do something more complicated
- Remember and repeat a bunch of commands
A program

- A program is a list of statements in a file
- x=2.0 is a statement
- Python executes the statements one by one
Comments

- Comments: start with the hash character, `#`, and extend to the end of the physical line
- Use to clarify code and are not interpreted by Python.
- Good programming practice.
- Examples:
  
  ```
  # this is the first comment
  spam = 1 # and this is the second comment.
  ```
Comments

- In the beginning, purpose of the program, structure, author, time,
- Before a statement or a group of statement, explain the purpose, the variables, the algorithm.
- For yourself, for others, for maintenance
- In our class, this will help us decide if you understand your program.
Your program 1

- Uses print (it is a function)
- Uses variables
- Uses remainder operator

- Please make sure your program runs!
Programming Lab Assignments

- Comments are **mandatory** – no comments, no points.
- Very similar programs
  - 50% off the first time
  - 75% off the second time
  - Student Judicial Affair the third time