ASSIGNMENT, LOOPS, AND BASIC TYPES

CSSE 120 – Rose-Hulman Institute of Technology

Outline (ch. 2, some of ch. 3)

- Identifiers
- Print statements
- Variables and assignments
- Definite loops
- Basic types: numbers (int and float)
- Math library
- Accumulator problem
- Back to the robots

Identifiers: Names in Programs

- Uses of *identifiers* so far…
  - Modules
  - Functions
  - Variables
- Rules for identifiers in Python
  - Start with a letter or _ (the "underscore character")
  - Followed by any sequence of letters, numbers, or _
- Case matters! spam ≠ Spam ≠ sPam ≠ SPAM
- Choose descriptive names!

Reserved Words

- Built-in names
- Can't use as regular identifiers
- Python reserved words:

and	del	for	is	raise
assert	elif	from	lambda	return
break	else	global	not	try
class	except	if	or	while
continue	exec	import	pass	with
def	finally	in	print	yield

Be careful not to redefine function names accidentally

- Examples:
  - len – used to find the number of items in a sequence
  - max
  - min
  - float – used to convert a number to a floating point number

Expressions

- Fragments of code that produce or calculate new data values
- Examples
  - *Literals*: indicate a specific value
  - *Identifiers*: evaluate to their assigned value
  - *Compound* expressions using *operators*: +, -, *, /, **
- Can use parentheses to group

Programming Languages

- Have precise rules for:
  - Syntax (form)
  - Semantics (meaning)
- Computer scientists use *meta-languages* to describe these rules
- Example…

Output Statements

- Syntax:
  - print
  - print <expr>
  - print <expr>, <expr>, …, <expr>,
  - print <expr>, <expr>, …, <expr>
- Semantics?
- Is this allowed?
  - print “The answer is:”, 7 * 3 * 2

A “slot” to be filled with any expression

Repeat indefinitely

Note: trailing comma

Variables and Assignments

- Variable
  - Identifier that stores a value
  - A value must be *assigned* to the variable
  - <variable> = <expr> (syntax)
- Assignment
  - Process of giving a value to a variable
  - Python uses = (equal sign)
    - x = 0.25
    - x = 3.9 * x * (1 – x)

Assignment Statements

1. Simple assignments
   - <variable> = <expr>
2. Input assignments
   - <variable> = input(<prompt>)
     - temp = input("Enter high temperature for today")
3. Compound assignments
   - <var>op=<expr> means <var> = <var> op <expr>
     where op is +, -, *, /,or %
     - Example: total += 5 is the same as total = total + 5
4. Simultaneous assignments
   - <var>, <var>, …, <var> = <expr>, <expr>, …, <expr>
     - sum, diff = x + y, x - y

Sequences

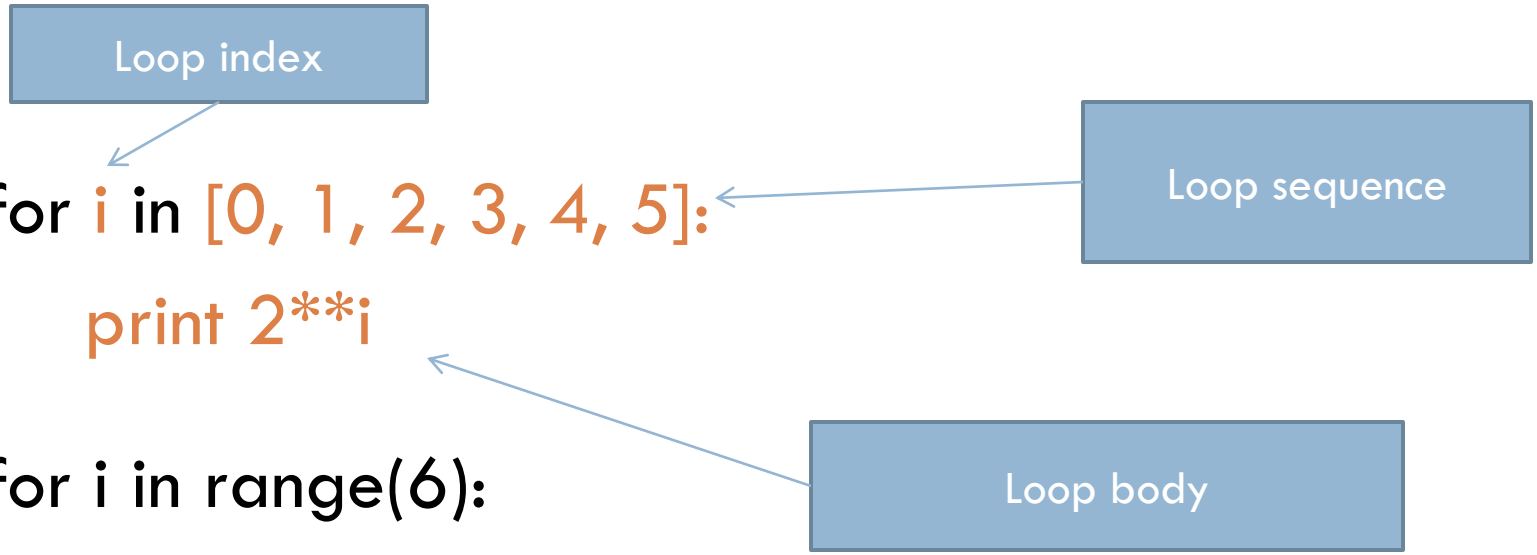
- A list of things
- For example:
  - [2, 3, 5, 7]
  - [“My”, “dog”, “has”, “fleas”]
- Some can be generated by the **range** function:
  - range(<expr>)
  - range(<expr>, <expr>)
  - range(<expr>, <expr>, <expr>)

Definite loops

- Definition
  - Loop: a control structure for executing a portion of a program multiple times
  - Definite: Python **knows** how many times to iterate the body of the loop
- Syntax:

```
for <var> in <sequence> :
    <body>
```

Examples using loops

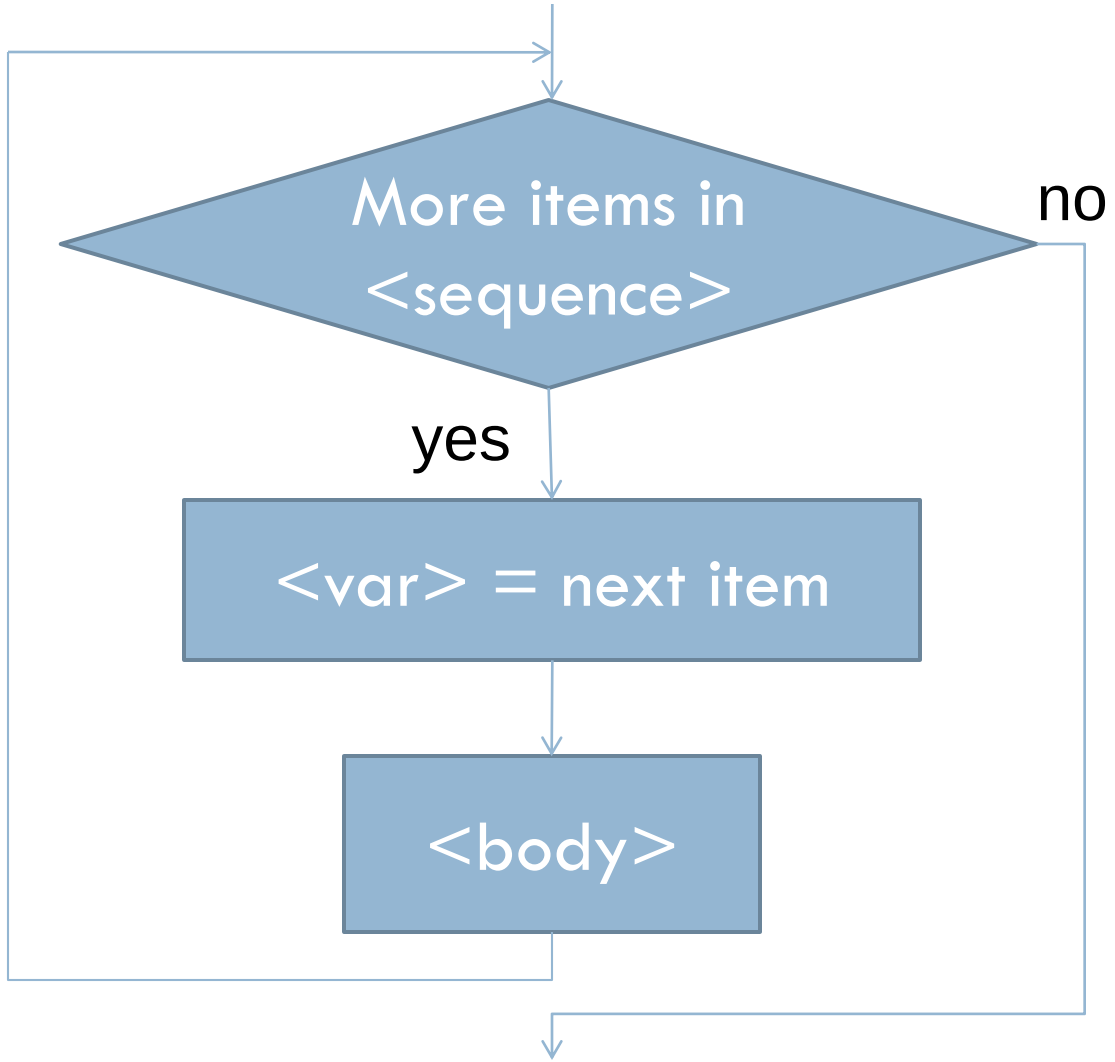



```
>>> for i in [0, 1, 2, 3, 4, 5]:
        print 2**i

>>> for i in range(6):
        print 2**i

>>> for b in ["John", "Paul", "George", "Ringo"]:
        print b, " was a Beatle"
```

Flowchart for a **for** loop



Trace this by hand:

```
a = 0
for i in range(4):
    a = a + i

print a
```

An ***accumulator*** combines parts of a list using looping.

We'll use this idea often this term!

Data types

- *Data*
  - Information stored and manipulated on a computer
  - Different kinds of data will be stored and manipulated in different ways
- *Data type*
  - A particular way of interpreting bits
  - Determines the possible values an item can have
  - Determines the operations supported on items

Numeric data types

```
print "Please enter the count of kind of coin."
quarters = input("Quarters: ")
dimes = input("Dimes: ")
nickels = input("Nickels: ")
pennies = input("Pennies: ")
total = quarters * 0.25 + dimes * 0.10 + nickels *
   .05 + pennies * .01
print "The total value of your change is", total
```

Finding the Type of Data

- Built-in function *type(<expr>)* returns the data type of any value
- Find the types of: 3, 3.0, -32, 64.0, “Shrubbery”, [2, 3]
- Why do we need different numerical types?
  - Operations on int are more efficient
    - Compute algorithm for operations on int are simple and fast
  - Counting requires int
  - Floats provide approximate values when we need real numbers

Built-in Help

- dir()
- dir(<identifier>)
- help(<identifier>)

- To see which functions are built-in, type:
  - dir(__builtins__)
- To see how to use them, type:
  - help(__builtins__)

Some Numeric Operations

Operator	Operation
+	Addition
-	Subtraction
*	Multiplication
/	Division
**	Exponentiation
%	Remainder
//	Do integer division (even on floats)

Function	Operation
abs(x)	Absolute value of x
round(x, y)	Round x to y decimal places
int(x)	Convert x to the int data type
float(x)	Convert x to the float data type

Q12b

Math library functions

Quadratic formula to find real roots for quadratic equations of the form $ax^2 + bx + c = 0$

- Solution:

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \qquad x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

- Write out the Python expression for the first formula.
- If you have time, test it IDLE

More math library components

Python	Mathematics	English
pi	π	Approximation of pi
e	e	Approximation of e
sin(x)	sin x	The sine of x
cos(x)	cos x	The cosine of x
tan(x)	tan x	The tangent of x
atan2(y,x)	$\tan^{-1}$ (y,x)	Arc tangent (inverse tangent) computes the angle formed by the positive x-axis and the ray from (0,0) to (x,y)
log(x)	ln x	The natural (base e) log of x
log10(x)	$\log_{10}x$	The base 10 log of x
exp(x)	e^x	The exponential of x

EXPLORING WITH PYTHON

Pair Programming

- Working in pairs on a single computer
  - One person, the *driver*, uses the keyboard
  - The other person, the *navigator*, watches, thinks, and takes notes
- For hard (or new) problems, this technique
  - Reduces number of errors
  - Saves time in the long run
- Works best when partners have similar skill level
- If not, then student with most experience should navigate, while the other student drives.

Food tasting

- Suppose you are at food tasting show and are tasting 5 different dishes
- Sampling the dishes in different orders may affect how good they taste
- If you want to try out every possible ordering, how many different orders would there be?
  - That number is the factorial of 5
  - $n! = n\,(n-1)\,(n-2)\,\ldots\,(1)$
- What type of problem is this?

Accumulating results: factorial

- Work in groups of two
  - Pick a driver and navigator
- Write a Python program that
  - Prompts the user for an integer
  - Calculates the factorial of the integer
    - $n! = n (n - 1) (n - 2) \ldots (1)$
  - Outputs the result to the screen
- Driver: email the code to your partner (so each has the program for the open-computer parts of exams)
- Submit one copy of program with both student's names in a program comment.
- Submit it in ANGEL to the Lessons > Homework > Homework 3 > Factorial Drop Box