CS108 Lecture 11:
Decision Making

Aaron Stevens
26 February 2008

Overview/Questions

– What’s the point?
– Logical Expressions
– If statements
– Complex logic (if/else if/else)
Decisions: what’s the point?

Previously, all of our programming code has always been executed – at least once.

We’ve used some techniques to modify this:
– Definite loops execute a fixed number of times.
– Functions allow us to execute the same code an arbitrary number of times.

Flow of Control

Sequential Execution
Each instruction is executed in order they are written (after the previous one, before the next one).

Functions
Functions modify the flow of control by delegating some work to be done by a separate functions, library, etc.
Flow of Control

Selection
Some statements are executed while others are not.

Repetition
Statements can be repeated some fixed number of time, or else can be repeated until some event signals they should not be repeated any more.

Logical Expressions

Expression
A programming statement which achieves a value. Python example: \(5 - 3\)

Logical Expression
Programming statement which achieves a Boolean logical result (True or False).
Python Logical Operators

Boolean constants
Python provides 2 boolean constants:
True – logical value, numeric value of 0
False – logical value, numeric value of not 0

Python Equality Operators

Equality
True if and only if both operands have the same value. Example: 5 == 3

Inequality
True if and only if the operands do not have the same value. Example: 5 != 3
Python Relational Operators

Less than  $<$
True if and only if LHS operand has a numeric value less than RHS operand.
Example: $5 < 3$

Greater than $>$
True if and only if LHS operand has a numeric value more than RHS operand.
Example: $5 > 3$

Python Relational Operators

Less than or equal $\leq$
True if LHS operand has a numeric value less than or equal to RHS operand.
Example: $5 \leq 3$

Greater than or equal $\geq$
True if LHS operand has a numeric value more or equal to RHS operand.
Example: $5 \geq 3$
Python Logical Operators

Logical AND

and
True if and only if both operands are True.
Example: \((3 < 5) \text{ and } (5 \geq 4)\)

Logical OR

or
True if either operands is True.
Example: \((3 < 5) \text{ or } (5 < 4)\)

Python Logical Operators

Logical Negation

not
True if operand has a value of False.
Example: \(\text{not } 0, \text{ not } True\)
if statement

A structure which evaluates a logical expression, and controls a block of statements.
grade = input("Enter a numeric grade: ")
if grade > 90:
    print "outstanding -- A"

Controlled Block

A set of statements which are “controlled” by a control statement. A controlled block must be indented:
grade = input("Enter a numeric grade: ")
if grade > 90:
    print "outstanding -- A"
else statement

else statement A control statement which is subordinate to a an if statement.

```python
grade = input("Enter a numeric grade: ")
if grade > 90:
    print "outstanding -- A"
else:
    print "better luck next time"
```

elif statement

elif statement A control statement which is subordinate to a an if statement, but checks a secondary condition. An elif is only evaluated if the first if evaluated to false.

```python
grade = input("Enter a numeric grade: ")
if grade > 90:
    print "outstanding -- A"
elif grade > 80:
    print "pretty good -- B"
else:
    print "better luck next time"
```
Variable Scope

Scope refers to the places in a program where a variable name can be referenced.

– Any variables introduced in a control structure are local to that structure – not accessible outside the block.
– Variables from outside the control structure are still accessible.

Nested logic

All control statements can be nested inside other control statements. Thus, some logical tests depend on the outcome of previous logical tests before even being evaluated.

Example: what clothing to wear based on temperature, conditions.
– 78 degrees and sun
– 78 degrees and rain
– 30 degrees and sun
– 30 degrees and sleet
Summary

– Decision making allows us to control the flow of execution within our programs.
– Logical expressions are evaluated to a Boolean result. This result determines the outcome of the control structure.
– Control structures can be combined to create any level of complex logic, using nesting – one control structure controlled by another.

Student To Dos

– Reading: 7.1-7.3 (today)
– Reading: 7.4-7.5, 8.1-8.3 (Thursday)
– Lab this week will be on decision making and repetition.
– HW 05 due tonight