Loops and
Conditionals
HORT 59000
Lecture 11
Instructor: Kranthi Varala

Relational Operators

• These operators compare the value of two ‘expressions’ and returns a Boolean value.

• Beware of comparing across data types, especially when reading values in from command line or files.

Relational Operators

==

equal

True if expressions are equal

!=

not equal

True if expressions are not equal

>

Greater than

True if left is greater than the right

<

Less than

True if left is less than the right

>=

greater than OR equal

<=

less than OR equal

is

identity

True if the left is the same object as right

General Python Syntax rules

• End of line is end of statement
• Statements at the same indentation level are in the same block (e.g., within a loop or condition)
• End of indentation
• Exceptions:
• Semi colon ; separates statements on the same line

• Single line blocks are allowed without indentation

Branching logic
• Used to implement alternate paths for the logic flow.

https://upload.wikimedia.org/wikipedia/commons/4/44/LampFlowchart.png

If/elif/else statements

```
if test1:
statement 1

elif test2:
statement 2

else:
statement 3
```

• Both the elif and else blocks are optional.

If/elif/else statements

Lamp flowchart with if/else

Accepts input from user, as a string

Example while loops

Altering while loops
• Normal loop control will execute all statements in block on every iteration. Loop ends only when exit condition is met.
• break statement forces the current loop to exit.
• continue statement skips the rest of the block and goes to the next iteration of the loop.
• pass statement is a placeholder for empty blocks.

Altering while loops

for loops
• for item in sequence:

```
statement 1
statement 2
.
```

.

• Generic iterator for items in a ordered sequence such as lists, tuples etc.
• On each iteration retrieves one item from the list and assigns it to the variable specified.
• Automatically moves to the next item in the order.
• Value of variable may be altered within the for loop, but change is not made in the list.

for loops

Looping over Strings and Lists
• List is a general sequence object while String is a character sequence object.
• Both can be iterated over by a for loop:

Looping over lists with and without index
• Looping with an index allows accessing the item within the list and changing it.

Looping over Tuples and Dictionaries

Nested Loops
• Loops can be nested just like the if/else statements.
• Indentation is again the key to creating nested loops.
• In a 2 level nested loop with x iterations on the outer loop and y iterations in the inner loop:
• All statements in the outer loop will be executed x times
• All statements in the inner loop will be executed x*y times

MultiDimensional Lists
my2DList = [[1,2,3,4],[5,6,7,8],[9,10,11,12],[13,14,15,16]]

0

1

2

3

0

1

5

9

13

1

2

6

10

14

2

3

7

11

15

3

4

8

12

16

my2DList