Loops in Python (part 1)

L555
Dept. of Linguistics, Indiana University
Fall 2010
Loops

Iteration
Loops allow us to carry out one or more statements repeatedly.

Syntax:

```python
while <test>:
    do ...
    do ...
```
While

Example

```python
mycount = 1
while mycount <= 100:
    print(mycount)
    mycount += 1
```

- the test is executed first, and if it is positive, the body of the loop is executed
  - the body of the loop is executed until the test is false
- if the test is false, the next statement after the loop is executed
  - if the test is false form the beginning, the body of the loop is not executed at all
Beware of infinite loops!
It is easy to create infinite loops with `while`. Make sure that your `while` condition will return false at some point.
Common Errors

Forgetting to advance your control variable
Make sure that the variable involved in the test changes in
the body of the loop.

Example

```python
mycount = 1
while mycount <= 100:
    print mycount
```
Common Errors

Wrong incrementing
If you want to decrement, make sure that you do not automatically increment

Example

```python
mycount = 100
while mycount > 0:
    print mycount
    mycount += 1
```
Common Errors

Off-by-1 errors

Getting the beginning and the end of a loop right can be tricky: Should it start with 0 or 1? And remember that the control variable is incremented before the loop is finished.

Example

```python
print 'multiples of 33'
mycount = 0
while mycount <= 100:
    mycount += 33
    print mycount
print 'This is the end: ' + str(mycount)
```
Use #1: iteration

As we’ve just seen, **while** loops can be used to **iterate** over a sequence.

- Commonly done by iterating over integers: integers easily count how many times you do something.
- You can change the way you iterate: \( i += 2, i -= 1, \ldots \)
Another, subtly different use is to perform the same actions until a certain condition is reached.

Example

```python
user_input = ""
while len(user_input) < 3:
    user_input = raw_input('Please enter a long string: ')
print "Thank you for entering a long enough string!"
```