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

Syntax:
while <test>:
do ...
do ...

While

Example
mycount = 1
while mycount <= 100:
    print mycount
    mycount += 1

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

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

Example
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`, ...

**Use #2: until**

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 long string: ')
print("Thank you for entering a long enough string!")
```