Introducing Repetition

How can we introduce error checking?

- Suppose that students are entering numbers into your miles per gallon program that aren’t valid

How can we introduce error checking?

- Suppose that students are entering numbers into your miles per gallon program that aren’t valid
  - Values not above 0
  - Ending value that is “smaller” than the starting value
  - Maybe even values above some upper limit
- How can we correct them and ask them to try again?

Repeating Statements

- Besides selecting which statements to execute, a fundamental need in a program is repetition
    - repeat a set of statements under some conditions
- Between selection and repetition, we have the two most necessary programming statements

While and For Statements

- The while statement is the more general repetition construct. It repeats a set of statements while some condition is True.
  - Often called a **sentinel controlled loop**
- The for statement is useful for iteration, moving through all the elements of data structure, one at a time.
  - Often called a **count controlled loop**

while Loop

- Top-tested loop (pretest)
  - test the boolean before running
  - Run the program suite
  - test the boolean before each iteration of the loop

```
while boolean expression:
    statementSuite
```

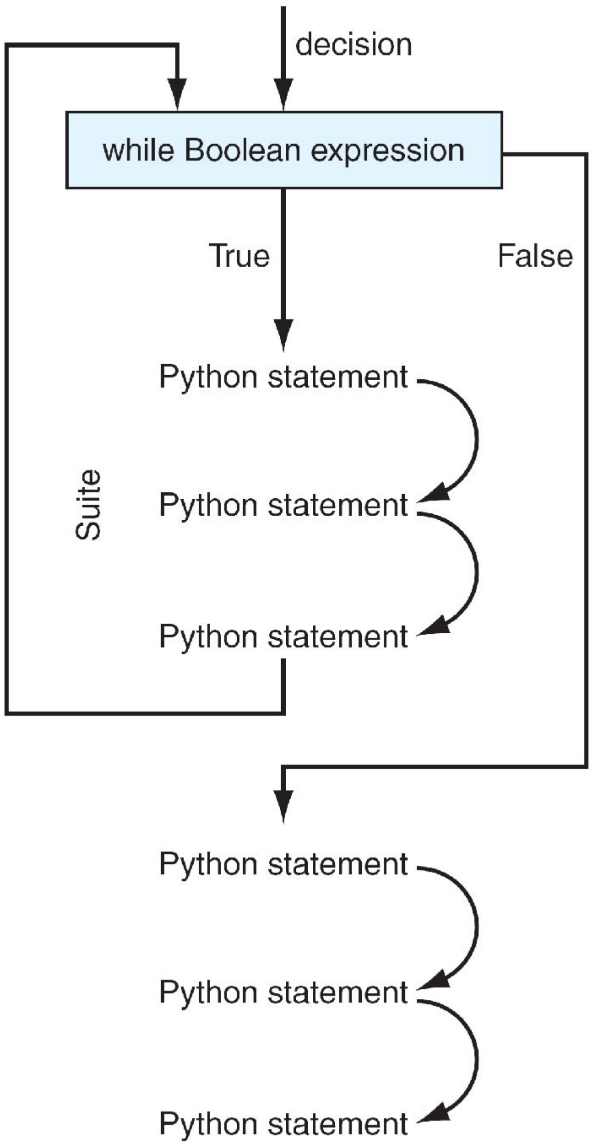

FIGURE 2.3 ***while*** loop.

Repeat While the Boolean is True

- while loop will repeat the statements in the suite while the boolean expression is True
- If the boolean expression never changes during the course of the loop, the loop will continue forever.

While Loop Example

```
x_int = 0

while x_int < 10:
    print (x_int)
    x_int = x_int + 1

print()
print( "Final value of x_int: ", x_int)
```

What is the Final Value printed by this code?

General Approach to a While

- outside the loop, initialize the boolean
- somewhere inside the loop you perform some operation which changes the state of the program,
  - eventually leading to a False boolean and exiting the loop
- Have to have both!

For and Iteration

- One of Python’s strength’s is it’s rich set of built-in data structures. Recall from ch. 3:
  - strings
  - lists and tuples
  - dictionaries
- The for statement is a common statement for manipulation of a data structure
  - for each element in the datastructure
    - perform some operation on that element

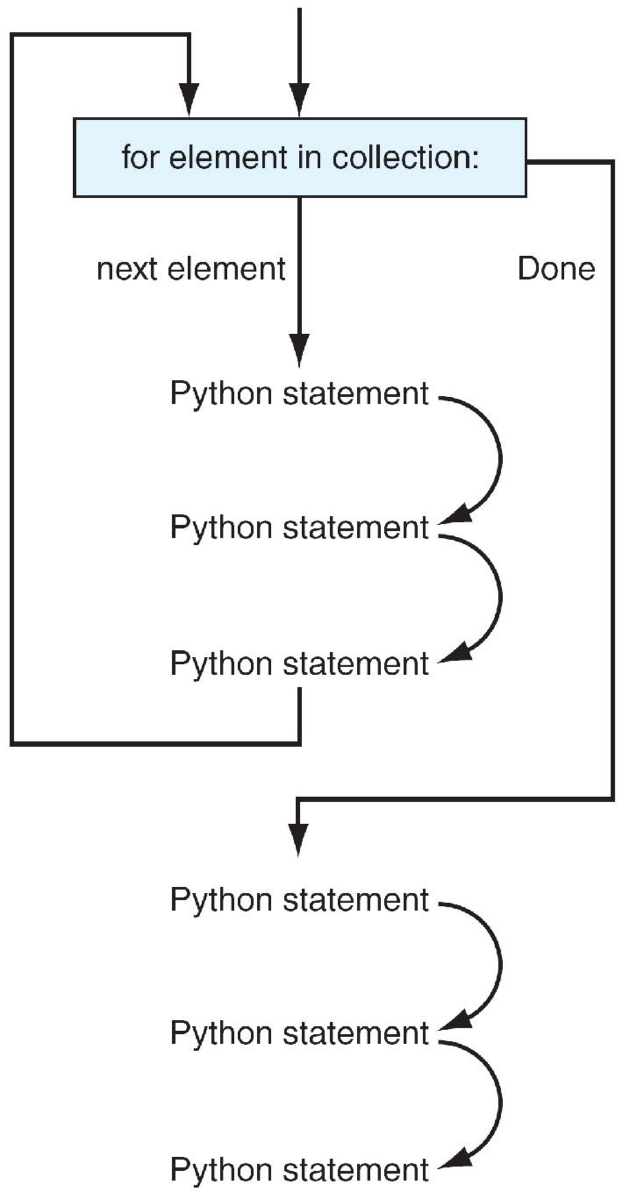

FIGURE 2.4 Operation of a `for` loop.

For Loop Example

```
numbers = [0,1,2,3,4,5,6,7,8,9]
for xInt in numbers:
    print (xInt)

print()
print ("Final value of xInt: " + str(xInt) )
```

Practice: so, how can we introduce error checking?

- Suppose that students are entering numbers into your miles per gallon program that aren’t valid
  - Values not above 0
  - Ending value that is “smaller” than the starting value
  - Maybe even values above some upper limit
- How can we correct them and ask them to try again?