3.7 Loop Patterns

Input Loops

Both of the following patterns read strings until they get a blank input.

```
done = False
while not done:
    x = input("Enter something: " )
    if x == "":
        done = True
    else:
        <whatever you want to do with x>
```

or

```
while True:
    x = input("Enter something: " )
    if x == "":
        break
    <whatever you want to do with x>
```

Process elements of a sequence

```
for x in <sequence>:
    <whatever you want to do with x>
```

or

```
x = <first element of the sequence>
while <x is in the sequence>:
    <whatever you want to do with x>
    x = <next element of the sequence>
```

The second version usually involves using a numeric index to refer to the first element of the sequence (index 0), the second element (index 1) and so forth. Where it makes sense the first version is usually easier.

Do something until a condition is satisfied

```
while not <condition>:
    <something>
```

or

```
while True:
    if <condition>:
        break
    something
```

This is equivalent to the first version; it uses a **break**-statement to exit from the loop

or

```
done = False
while not done:
    <something>
    if <condition>:
        done = True
```

This allows the <condition> to be created by the action <something>.

or

```
while True:
    <something>
    if <condition>:
        break
```

This is completely equivalent to the preceding version; it just uses a **break**-statement to exit from the loop. Note that this differs from the second version only in the other of the statements.