Announcements

- Homework 3 due Tuesday 10/1 @ 11:59pm
- Optional Hog Contest due Thursday 10/3 @ 11:59pm
- Homework 4 due Tuesday 10/8 @ 11:59pm
- Project 2 due Thursday 10/10 @ 11:59pm
- Guerrilla Section 2 this Saturday 10/5 & Sunday 10/6 10am-1pm in Soda
  - Topics: Data abstraction, sequences, non-local assignment
  - Meet outside Soda 306

Sequence Iteration

**For Statements**

```python
def count(s, value):
    total = 0
    for element in s:
        if element == value:
            total = total + 1
    return total
```

(Demo)

**For Statement Execution Procedure**

```python
for <name> in <expression>:
    <suite>
```

1. Evaluate the header `<expression>`, which must yield an iterable value (a sequence).
2. For each element in that sequence, in order:
   A. Bind `<name>` to that element in the first frame of the current environment.
   B. Execute the `<suite>`.

**Sequence Unpacking in For Statements**

```python
>>> pairs = (((1, 2), (2, 2), (2, 3), (4, 4)))
>>> same_count = 0
```

```python
>>> for x, y in pairs:
        if x == y:
            same_count = same_count + 1
```

```python
>>> same_count
2
```
Ranges

The Range Type

A range is a sequence of consecutive integers:

..., -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5 ...

Length: ending value - starting value
Element selection: starting value + index

Tuple constructor
With a 0 starting value

"Ranges can actually represent more general integer sequences."
Limitations on Dictionaries

Dictionaries are unordered collections of key-value pairs.

Dictionary keys do have two restrictions:

• A key of a dictionary cannot be an object of a mutable built-in type.

• Two keys cannot be equal. There can be at most one value for a given key.

This first restriction is tied to Python's underlying implementation of dictionaries.

The second restriction is an intentional consequence of the dictionary abstraction.

If you want to associate multiple values with a key, store them all in a sequence.