# CSCI 1300

## Intro to Computing

Gabe Johnson

Lecture 12 Feb 11, 2013

# Associative Arrays
# How Programs Work

# Upcoming Homework Assignment

HW #3 **Due: Friday, Feb 15**

## Lists and Dictionaries

This week's homework is all about using *lists* and a newly introduced data structure called *associative arrays*, a.k.a. *dictionaries*. HW will be out later today.

# Announcements

1. **Tests** are graded but not entered into RG yet. I'll tweet when they are up and you can log in and see how you did. With 50 graded, avg is 23. Big variance.

2. **RetroGrade Accounts** for 24 people (this class and the other) had to be merged because there were two. If you can't log in, email me *with your student id and email addresses you might have used*.

# Announcements

3. **Advanced Bash and Scripting** session this Wednesday night. 7:30pm in ECCR 105.

# Dictionaries

Dictionaries. Associative Arrays. Hashes. Maps.

All these refer to the same thing. In Python, they are called Dictionaries, though.  Here’s a rad one:

```
dog = { 'name' : 'Sputnik',
        'pronoun' : 'he',
        'pos_pron' : 'his',
        'age' : 4,
        'occupation' : 'Being Awesome',
}
```

# Dictionaries = Awesome

Dictionaries let you group together a bunch of data that are related in some way. They are one stepping stone to doing object-oriented programming.

# Anatomy of a Dictionary

```
# create an associative array like this:
dog = { 'name' : 'Sputnik', # dog's name
        'pronoun' : 'he',   # pronoun: he/she/it
        'pos_pron' : 'his', # possessive pronoun: his/her/its
        'age' : 4,          # dog's age (an integer)
        'occupation' : 'Being Awesome', # note last comma is ok
}
```

‘dog’ is the name of this associative array. We can initialize this dictionary by assigning something that looks like the right side of the equals sign.

# Anatomy of a Dictionary

```
# create an associative array like this:
dog = { 'name' : 'Sputnik', # dog's name
        'pronoun' : 'he',   # pronoun: he/she/it
        'pos_pron' : 'his', # possessive pronoun: his/her/its
        'age' : 4,          # dog's age (an integer)
        'occupation' : 'Being Awesome', # note last comma is ok
}
```

To tell Python this is a dictionary and not a list or a string or an integer, use the curly braces, and put some key/value pairs on the inside, separated by commas.

# Anatomy of a Dictionary

```
# create an associative array like this:
dog = { 'name' : 'Sputnik', # dog's name
        'pronoun' : 'he',   # pronoun: he/she/it
        'pos_pron' : 'his', # possessive pronoun: his/her/its
        'age' : 4,          # dog's age (an integer)
        'occupation' : 'Being Awesome', # note last comma is ok
}
```

Key/value pairs have the following format:

```
   some_key : some_value
```

# Anatomy of a Dictionary

```
# create an associative array like this:
dog = { 'name' : 'Sputnik', # dog's name
        'pronoun' : 'he',   # pronoun: he/she/it
        'pos_pron' : 'his', # possessive pronoun: his/her/its
        'age' : 4,          # dog's age (an integer)
        'occupation' : 'Being Awesome', # note last comma is ok
}
```

```
some_key : some_value
```

There are limitations to what keys and values can be.

For our purposes, strings and integers will work.

# Anatomy of a Dictionary

```
# create an associative array like this:
dog = { 'name' : 'Sputnik', # dog's name
        'pronoun' : 'he',   # pronoun: he/she/it
        'pos_pron' : 'his', # possessive pronoun: his/her/its
        'age' : 4,          # dog's age (an integer)
        'occupation' : 'Being Awesome', # note last comma is ok
}
```

```
    some_key1 : some_value1,
    some_key2 : some_value2,
         < and so on>
```

Separate key and values with a colon.

Separate key/value pairs with commas.

# Using Dictionaries

```
# now you can use the values contained in this dictionary by
# using the 'dog' variable, the square brackets, and the key
# that you're interested in using. Like this:
print "My dog's name is", dog['name']
print dog['pos_pron'], "occupation is", dog['occupation']
print dog['pronoun'], "is", dog['age'], "years old."
```

# Using Dictionaries

```
# now you can use the values contained in this dictionary by
# using the 'dog' variable, the square brackets, and the key
# that you're interested in using. Like this:
print "My dog's name is", dog['name']
print dog['pos_pron'], "occupation is", dog['occupation']
print dog['pronoun'], "is", dog['age'], "years old."
```

This prints:

```
My dog's name is Sputnik
his occupation is Being Awesome
he is 4 years old.
```

# Enough! Let’s code.