



CSCI 1300

Intro to Computing

Gabe Johnson

Lecture 16

Feb 22, 2013

Objects: Instance Methods

Lecture Goals

1. Special Announcement BA/CS
2. Objects and Methods
3. Persistent State

B.A. In CS is Open!

I will forward an email from Ken Anderson later today about the new Bachelor of Arts in Computer Science. You can sign up at the A&S Majors fair, which is rumored to be today (I don't know anything beyond that, so don't ask :)

The email will have details.

More Methods!

Below, 'bark' is a method:

```
class Dog:
    name = "Sparky"
    def bark(self):
        print "Woof"
```

But 'wag' is not a method. Why?

```
def wag(dog):
    print dog.name + " wags the tail"
```

Methods are part of Classes

Methods are:

1. Defined inside a class
2. Accept a 'self' argument in the first position.

Dot Notation

To call a method 'bark' on an instance of the 'Dog' class:

```
dog.bark()
```

Notice that even though the definition is

```
def bark(self)
```

... we *don't* pass anything when we make a method call.

Chaining Dots

We can have objects that contain other objects, and we can use the dot-notation on each of them.

```
class Dog:
    buddy = None
    def bark(self):
        print "woof"
```

```
d1 = Dog()
d2 = Dog()
d1.buddy = d2
d1.buddy.bark()
```

More Chaining

```
d1 = Dog()  
d2 = Dog()  
d3 = Dog()  
d4 = Dog()  
d5 = Dog()  
d1.buddy = d2  
d1.buddy.buddy = d3  
d1.buddy.buddy.buddy = d4  
d1.buddy.buddy.buddy.buddy = d5  
d1.buddy.buddy.buddy.buddy.bark()
```


Methods can return Objects

```
class Dog:
    age = 0
    fr1 = None
    fr2 = None
    def get_older(self):
        if self.fr1.age > self.fr2.age:
            return self.fr1
        else:
            return self.fr2
```

```
sputnik = Dog()
sputnik.fr1 = Dog()
sputnik.fr2 = Dog()
sputnik.fr1.age = 5
sputnik.fr2.age = 8
```

```
old = sputnik.get_older()
# old is now fr2
old.bark()
```