22.4. Invoking the Parent Class's Method

Sometimes the parent class has a useful method, but you just need to execute a little extra code when running the subclass's method. You can override the parent class's method in the subclass's method with the same name, but also invoke the parent class's method. Here's how.

Say you wanted the Dog subclass of Pet to say "Arf! Thanks!" when the feed method is called, as well as executing the code in the original method.

Here's the original Pet class again.

```
Save & Run Original - 1 of 1 Show in CodeLens

1 from random import randrange
2
3   # Here's the original Pet class
4 class Pet():
5   boredom_decrement = 4
6   hunger_decrement = 6
7   boredom_threshold = 5
8   hunger_threshold = 10
9   sounds = ['Mrrp']
10   def __init__(self, name = "Kitty"):
11   self.name = name
12   self.name = name
13   self.boredom = randrange(self.hunger_threshold)
14   self.sounds = self.sounds[:]   # copy the class attribute, so that when we number of the self.sounds = self.sounds[:]  # copy the class attribute, so that when we number of the self.sounds = self.sounds[:] # copy the class attribute, so that when we number of the self.sounds[:] # copy the class attribute, so that when we number of the self.sounds[:] # copy the class attribute, so that when we number of the self.sounds[:] # copy the class attribute, so that when we number of the self.sounds[:] # copy the class attribute, so that when we number of the self.sounds[:] # copy the class attribute, so that when we number of the self.sounds[:] # copy the class attribute, so that when we number of the self.sounds[:] # copy the class attribute, so that when we number of the self.sounds[:] # copy the class attribute, so that when we number of the self.sounds[:] # copy the class_copy)
```

And here's a subclass that overrides feed() by invoking the the parent class's feed() method; it then also executes an extra line of code. Note the somewhat inelegant way of invoking the parent class' method. We explicitly refer to Pet.feed to get the method/function object. We invoke it with parentheses. However, since we are not invoking the method the normal way, with <obj>.methodname, we have to explicitly pass an instance as the first parameter. In this case, the variable self in Dog.feed() will be bound to an instance of Dog, and so we can just pass self: Pet.feed(self).

```
Save & Run Original - 1 of 1 Show in CodeLens

1 from random import randrange
2
3 class Dog(Pet):
4 sounds = ['Woof', 'Ruff']
5
6 def feed(self):
7 Pet.feed(self)
8 print("Arf! Thanks!")
9
10 dl = Dog("Astro")
11
12 dl.feed()
13

Arf! Thanks!

Activity: 2 -- ActiveCode (feed_me_example)
```

Note

There's a better way to invoke a superclass's method. Unfortunately, the implementation of python in our ActiveCode windows doesn't support it, so we aren't using it here. In that alternative method, we would call super().feed(). This is nice because it's easier to read, and also because it puts the specification of the class that Dog inherits from in just one place, class Dog(Pet). Elsewhere, you just refer to super() and python takes care of looking up that the parent (super) class of Dog is Pet.

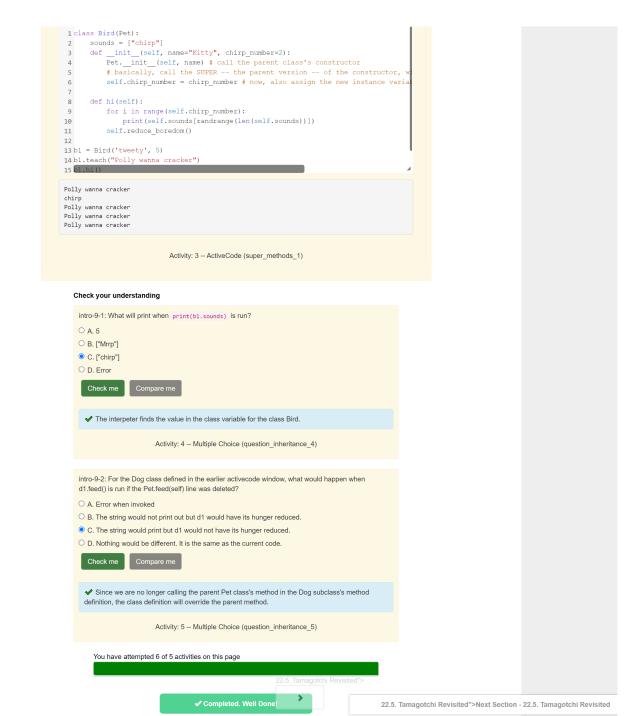
This technique is very often used with the init method for a subclass. Suppose that some extra instance variables are defined for the subclass. When you invoke the constructor, you pass all the regular parameters for the parent class, plus the extra ones for the subclass. The subclass init method then stores the extra parameters in instance variables and calls the parent class init method to store the common parameters in instance variables and do any other initialization that it normally does.

Let's say we want to create a subclass of Pet, called Bird, and we want it to take an extra parameter, chirp_number, with a default value of 2, and have an extra instance variable, self.chirp_number. Then, we'll use this in the hi method to make more than one sound.

Save & Run

Original - 1 of 1

Show in CodeLens



22.3. Overriding Methods">

rerriding Methods">

© Copyright 2017 bradleymiller. Created using Runestone 4.1.17.

| Back to top