

## Re: why is "self" used in OO-Python?

---

*Source:* <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2008-07/msg01211.html>

---

- *From:* castironpi <castironpi@xxxxxxxx>
  - *Date:* Sat, 12 Jul 2008 11:44:40 -0700 (PDT)
- 

On Jul 12, 1:01 pm, Duncan Booth <duncan.bo...@xxxxxxxxxxxxxxxx> wrote:

ssecorp <circularf...@xxxxxxxx> wrote:

1. Why do I have to pass self into every method in a class? Since I am always doing why cant this be automated or abstracted away?  
Are the instances where I won't pass self?  
I imagine there is some tradeoff involved otherwise it would have been done away with.

When you define a method in Java there is an implicit 'this' passed to the method. Python cannot tell when you define a function whether the function is going to be used as a function, an instance method, a class method, a static method or something else (or all of the above). Consider this:

The dynamic nature of Python means you can lift a method out of a class and re-use it in a different context or inject a function into a class as a method. There are two ways to handle this sort of code: javascript has an implied 'this' for everything whether a function or what passes for a method, Python makes it explicit.

2. self.item instead of getters and setters. I thought one of the main purposes of OO was encapsulation. Doesn't messing with internal object-representations break this?

That is correct. Some languages (e.g. Java) don't allow you to encapsulate attributes so you have to write getter and setter methods. If you expose an attribute in Java then you cannot later insert some code into the lookup or override the set without getting all users of your code to change the way they access the value. This is bad.

Other languages (e.g. Python, C#) allow you to intercept the attribute lookup so you can change a plain attribute into a property without requiring the users of your class alter their source code. With C# I think they would still need to recompile their code so it may be more appropriate to avoid using public attributes if you are producing a class

Re: why is "self" used in OO-Python?

library for widespread reuse, but with Python there is no difference to the user of your class whether they are accessing an attribute or a property.

Sadly a lot of Java programmers mistake the limitations of their language for r