

# understaning "self"

---

*Source:* <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2008-02/msg02761.html>

---

- *From:* "Poppy" <[znfmail-pythonlang@xxxxxxxxxx](mailto:znfmail-pythonlang@xxxxxxxxxx)>
  - *Date:* Thu, 21 Feb 2008 08:34:25 -0500
- 

I've been searching online to try get a better understanding of what "self" does when I define this parameter in my class functions. All I'm finding is debates on whether "self" has any value to the language but that doesn't help me in my newbie question. So the code excerpt below is from "Beginning Python" Norton, Samuel, Aitel, Foster-Johnson, Richardson, Diamon, Parker, and Roberts.

What I think "self" is doing is limiting the function call to only function in "this" class. So in the function below "has" calls self.has\_various(), if I had a function called "has\_various" in my program or another included class using "self" insures that the "has\_various" below is the one used. Am I correct in my understanding?

thanks,

Zach-

```
def has(self, food_name, quantity=1):
    """
    has(food_name, [quantity]) - checks if the string food_name is in the
    fridge. quantity defaults to 1
    returns True if there is enough, false otherwise.
    """

    return self.has_various({food_name:quantity})

def has_various(self, foods):
    """
    has_various(foods) determines if the dictionary food_name
    has enough of every element to satisfy a request.
    returns true if there's enough, Faslse if there's not or if an element does
    not exist.
    """
    try:
        for food in foods.keys():
            if self.items[food] < foods[food]:
```

understaning "self"

```
return False  
return True  
except KeyError:  
return False
```

.