

Re: My fight with classes :)

Re: My fight with classes :)

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2008-06/msg01273.html>

- *From:* Peter Pearson <ppearson@xxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 11 Jun 2008 10:38:14 -0500
-

On Wed, 11 Jun 2008 22:16:56 +0800, TheSaint <fc14301589@xxxxxxxxxxxxx> wrote:

Hi,
I'm very new with classes. I still reading something around ;)

I got started to try a concatenation of 2 type of string, which have a particular property to start with A or D.

My class here:

```
""" Small class to join some strings according to the leading first letter """
```

```
def __init__(self):  
    self.valueA= "  
    self.valueD= "  
  
def __add__(self, value):  
    if not isinstance(value, str): return  
    if value.lower().startswith('a'):  
        self.valueA += value  
    if value.lower().startswith('d'):  
        self.valueD += value  
    return self.valueA ,self.valueD
```

```
__call__ = __add__  
__iadd__ = __add__
```

my test on the shell:

[snip]

```
                k +'aks'  
  
('aks', "  
  
                k +'daks'  
  
('aks', 'daks')
```

Re: My fight with classes :)

Re: My fight with classes :)

[snip]

```
k += 'liu'  
k += 'aliu'
```

```
Traceback (most recent call last):  
File "<stdin>", line 1, in <module>  
TypeError: can only concatenate tuple (not "str") to tuple
```

You have designed your class in a confusing way, and then you were confused by it.

It was confusing to design your class in such a way that "k+'aks'" modifies k, because "k+'aks'" appears to be just an expression. Changing k as a side-effect of evaluating that expression is unnecessarily confusing. See if you can figure out how to design your class so that you modify k either by writing "k.append('aks')" or "k += 'aks'".

Speaking of which, you discovered that "k += 'liu'; k += 'aliu'" fails. It fails because "k += 'liu'" replaces k with a tuple (because you return a tuple from your `__add__` function), so k is no longer an instance of your class.

Do I miss something?

I'd rather like to avoid class, but a function won't allow me to store so easily data between several call.

Classes are wonderfully useful, and it would be sad for you to use Python while shunning classes. I strongly recommend looking at some good examples of simple class programming; I apologize for having no specific recommendations.

--

To email me, substitute nowhere->spamcop, invalid->net.

.

Re: My fight with classes :)