Caching Design Issue #2
Mutable Objects

- Let's say we have a class that extends **`list`**
- Our extension computes sums of items in the list
- And caches the sums to save recomputing it all the time
- What happens when the list changes?

Extending a List

- How about this? Will this work?

```
class BadSumList(list):
    def __init__(self, *args, **kw):
        super().__init__(*args, **kw)
        self._sum = None
    def sum(self):
        if not self._sum:
            self._sum = sum(self)
        return self._sum
```