

python singleton

```
def singleton(cls):
    instances = [] # python2 nonlocalist
    def getinstance(*args, **kwargs):
        if not instances:
            instances.append(cls(*args, **kwargs))
        return instances[0]
    return getinstance

@singleton
class Foo:
    a = 1
```

```
class Singleton(object):
    def __new__(cls, *args, **kwargs):
        if not hasattr(cls, '_instance'):
            cls._instance = object.__new__(cls, *args, **kwargs)
        return cls._instance

class Foo(Singleton):
    a = 1
```

Metaclass

```
class Singleton(type):
    def __call__(cls, *args, **kwargs):
        if not hasattr(cls, '_instance'):
            cls._instance = super(Singleton, cls).__call__(*args, **kwargs)
        return cls._instance

class Foo():
    __metaclass__ = Singleton
```