

---

# **Aaron Documentation**

***Release 1.0.1***

**Brian Hicks**

January 05, 2013



# CONTENTS



Aaron adds some syntactic sugar for Python function composition.

Contents:



# API

Aaron adds some syntactic sugar for Python function composition.

## 1.1 Composition

```
class aaron.composition.Composition(*items)
    a single composition, possibly with other compositions inside it

    __call__(*args, **kwargs)
        Call has two basic steps: getting an initial value, then running that initial value through each step of the
        composition.

        Note that the initial function will always receive *args and **kwargs. All subsequent calls re-
        spect the second pair of the tuple, which will splat the output of the previous function (using the form
        func(*result).)

    __gt__(other)
        Use > to append to the composition, without splatting.

    __init__(*items)
        initialize this Composition by _wrapping each item in a list

    __rshift__(other)
        Use >> to append to the composition, with splatting.

    __weakref__
        list of weak references to the object (if defined)
```

## 1.2 Decorators

```
aaron.decorators.composable(func)
    return a Composition object with just this function
```





# INDICES AND TABLES

- *genindex*
- *modindex*
- *search*



# PYTHON MODULE INDEX

## a

aaron, ??

aaron.composition, ??

aaron.decorators, ??