

デコレータっていいよね

デコレータの紹介をします。

@が嫌で使ったことがないかたは
ぜひ試してみてください。

田原悠西

デコレータって何？

関数とメソッドに使えます。

@の次にcallableなオブジェクトの名前を書きます。

例 ↓

```
@property
def size(self):
    return len(self.data)
```

これと同じ意味 ↓

```
def size(self):
    return len(self.data)
size = property(size)
```

基本は合成関数

引数の関数を呼出す関数を返す関数

```
def double(func):
```

“引数の関数を二回適用する関数を返す”

```
def f(*args, **kw):
```

```
    return func(func(*args, **kw))
```

```
return f
```

使ってみましょう

```
>>> @double
>>> def inc(n):
>>>     return n+1
>>> inc(1)
3
```

```
>>> @double
>>> @double
>>> def inc(n):
>>>     return n+1
>>> inc(1)
4
```

具体例1

(What's New in Python 2.4より)

```
def require_int(func):  
    def wrapper(arg):  
        assert isinstance(arg, int)  
        return func(arg)  
    return wrapper
```

```
@require_int  
def p1(arg):  
    print arg
```

具体例2

簡単なキャッシュ機能

```
def cache(func):  
    data = {}  
    def f(*args):  
        if args in data:  
            return data[args]  
        result = func(*args)  
        data[args] = result  
        return result  
    return f
```

フレームワーク

フレームワークが提供するデコレータを使う場合が一番多いかもしれません。

(*zope.formlib*の場合)

```
class MyForm(form.EditForm):  
    @action('Save', validator='myvalidator')  
    def handle_save_action(self, action, data):  
        #ここに処理を書く....
```

さらに高度に使いこなすなら

合成関数では物足りない人は、デスクリプタ (descriptor) を使いましょう。

Objects/descrobject.cにあるpropertyのPython実装が参考になります。