



# 程序设计与数据结构

## —— 函数的高级应用

讲师：周宇航

1. 理解Python中函数即变量
2. Map函数
3. Reduce函数
4. 匿名函数
5. 装饰器

- 变量可以指向函数
- 函数名也是变量
- 传入函数
- 函数本身也可以赋值给变量，即：变量可以指向函数

- map()函数接收两个参数，一个是函数，一个是Iterable
- map将传入的函数依次作用到序列的每个元素，并把结果作为新的Iterator返回
- ```
>>> def f(x):  
...     return x * x  
...  
>>> r = map(f, [1, 2, 3, 4, 5, 6, 7])  
>>> list(r)  
[1, 4, 9, 16, 25, 36, 49]
```

- reduce把一个函数作用在一个序列[x1, x2, x3, ...]上，reduce把结果继续和序列的下一个元素做累积计算
- $\text{reduce}(f, [x1, x2, x3, x4]) = f(f(f(x1, x2), x3), x4)$
- 把序列[7,8,3,4,9,7]变换成整数783497  

```
>>> from functools import reduce  
>>> def fn(x, y):  
...     return x * 10 + y  
>>> reduce(fn, [...])
```

- 在代码运行期间动态增加功能的方式，称为 “装饰器” (Decorator)
- @property
- @setter

- 函数作为参数传递
- 匿名函数
- map/reduce函数
- 了解装饰器原理，面试热点问题
- 使用@property修饰对象方法变属性

- 利用map()函数，把用户输入的不规范的英文单词，变为首字母大写，其他小写的规范名字。输入：

[ 'MIKE' , 'Adidas' , ' coffee '], 输出：

[ ' Mike ' , ' Adidas ' , ' Coffee ']:

- # def normalize(name):

# pass 实现这个函数

- ## 测试:

```
# L1 = [ 'MIKE' , 'Adidas' , ' coffee']
```

```
# L2 = list(map(normalize, L1))
```

```
# print(L2)
```



- 请设计一个装饰器，它可以作用于任何函数上，打印函数执行时间：

```
def metric(fn):  
    print('%s executed in %s ms' % (fn.__name__, 10.24))  
    return fn
```

注意：一定要自己动手联系，此作业不用上交

# EDU

CSDN学院 IT实战派

