

人工智能实践教程

从Python入门到机器学习

- 所有代码及ppt均可以由以下链接下载
- <https://github.com/shao1chuan/pythonbook>
- <https://gitee.com/shao1chuan/pythonbook>

目录

CONTENT
S

1. 生成式与生成器
2. 迭代器
3. 闭包
4. 装饰器
5. 内置高阶函数

列表生成式就是一个用来生成列表的特定语法形式的表达式。

是Python提供的一种生成列表的简洁形式,可快速生成一个新的list。

普通的语法格式: `[exp for iter_var in iterable]`

带过滤功能语法格式: `[exp for iter_var in iterable if_exp]`

循环嵌套语法格式: `[exp for iter_var_A in iterable_A for iter_var_B in iterable_B]`

1. 生成一个 $2n+1$ 的数字列表， n 为从3到11的数字。
2. 求以 r 为半径的圆的面积和周长(r 的范围从1到10)。
3. 找出1~100之间所有的质数。
4. 找出1~100之间所有奇数，并且返回一个列表。如果能被3整除，返回返回该数的平方，否则返回该数的三次方。
5. 给定一个列表 $li = [2,3,4,5,6,3,1,4,5]$ ，如果是偶数返回0，如果是奇数返回1。

字典生成式：用来快速生成字典；

集合生成式：用来快速生成集合；

```
s = {1,2,3,4,5,6,7}
# 集合生成式
print({i**2 for i in s })
# 字典生成式
print({i:i**2 for i in s })
```

- 1). 将字典的key值和value值调换;
- 2). 大小写计数合并：已知字典{'A':10, 'b':5, 'a':2}, 合并后为{'a':12, 'b':5}
key值最终全部为小写.
- 3). 假设有20个学生，学生分数在60-100之间，筛选出成绩在90分以上的学生;

什么叫生成器？

在Python中,一边循环一边计算的机制,称为**生成器:Generator**。

什么时候需要使用生成器？

一般情况下我们不需要使用生成器,只有当我们因为性能限制才需要用到,比如我们使用python读取一个10g的文件,如果一次性将10g的文件加载到内存处理的话(read方法),内存肯定会溢出;这里如果可以使用生成器把读写交叉处理进行,比如使用(readline和readlines)就可以再循环读取的同时不断处理,这样就可以节省大量的内存空间.

如何创建生成器？

第一种方法: 列表生成式的改写。 []改成()

第二种方法: yield关键字。

如何打印生成器的每一个元素呢？

通过for循环， 依次计算并生成每一个元素。

如果要一个一个打印出来,可以通过next()函数获得生成器的下一个返回值。



生成器的特点是什么？

1. 节约内存
2. 迭代到下一次的调用时,所使用的参数都是第一次所保留下的,即是说,在整个所有函数调用的参数都是第一次所调用时保留的,而不是新创建的。

生成器的应用案例: 斐波那契数列

除第一个和第二个数外,任意一个数都可由前两个数相加得到.

1, 1, 2, 3, 5, 8, 13, 21, 34, ...

生成器的应用案例：聊天机器人

[用户粉条]>>: 你的姓名?

[机器人]>>: 我是聊天机器人Siri

[用户粉条]>>: 你吃饭了吗?

[机器人]>>: 你吃饭了

[用户粉条]>>: 今天天气好吗?

[机器人]>>: 今天天气好

[用户粉条]>>: q

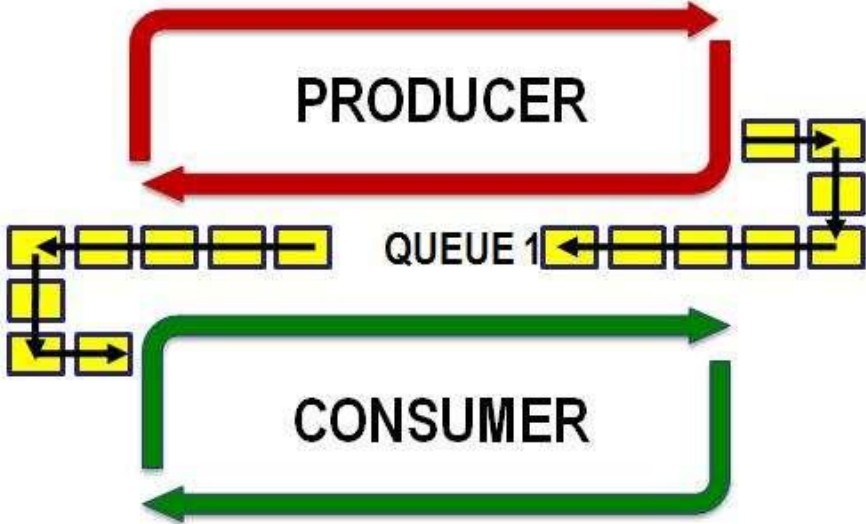
机器人累了，需要休息.....下次再聊



生成器的应用案例: 生产者-消费者模型

生产者:
采集数据

消费者:
分析数据



生成器的应用案例: 求平均值

```
请输入求平均值的数: 1  
[1] 的平均值为: 1.0  
请输入求平均值的数: 2  
[1, 2] 的平均值为: 1.5  
请输入求平均值的数: 3  
[1, 2, 3] 的平均值为: 2.0  
请输入求平均值的数: 4  
[1, 2, 3, 4] 的平均值为: 2.5  
请输入求平均值的数: q  
程序执行结束.....
```

迭代是访问容器元素的一种方式。迭代器是一个可以记住遍历的位置的对象。

迭代器对象从集合的第一个元素开始访问,直到所有的元素被访问完结束。

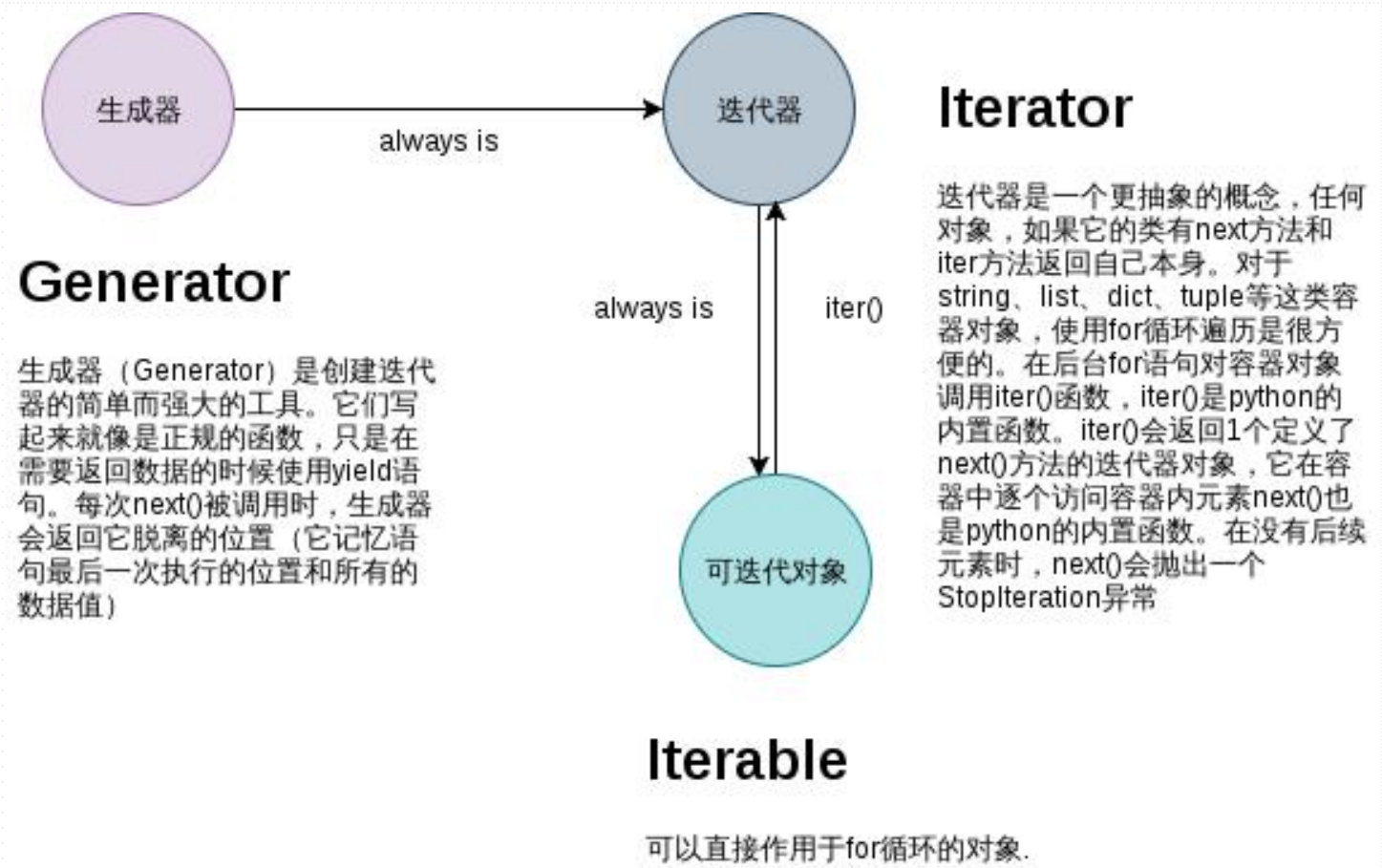
可迭代对象: 可以直接作用于for循环的对象(如何判断是否可以迭代?)

一类是集合数据类型,如list, tuple, dict, set, str等;

一类是generator,包括生成器和带yield的generator function。

- 可以被`next()`函数调用并不断返回下一个值的对象称为迭代器:`Iterator`。
- 生成器都是`Iterator`对象,但`list`、`dict`、`str`虽然是`Iterable`,却不是`Iterator`。
- 把`list`、`dict`、`str`等`Iterable`变成`Iterator`可以使用`iter()`函数

Python面试题: 迭代器和生成器的区别？



闭包

函数引用?

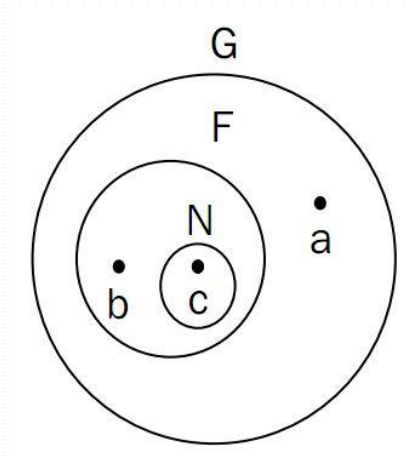
什么是闭包?

闭包的概念就是当我们在函数内定义一个函数时，这个内部函数使用了外部函数的临时变量，且外部函数的返回值是内部函数的引用时，我们称之为闭包。

闭包再理解? 内部函数对外部函数作用域里变量的引用(非全局变量),则称内部函数为闭包。

nonlocal关键字? 显式的指定变量不是闭包的局部变量

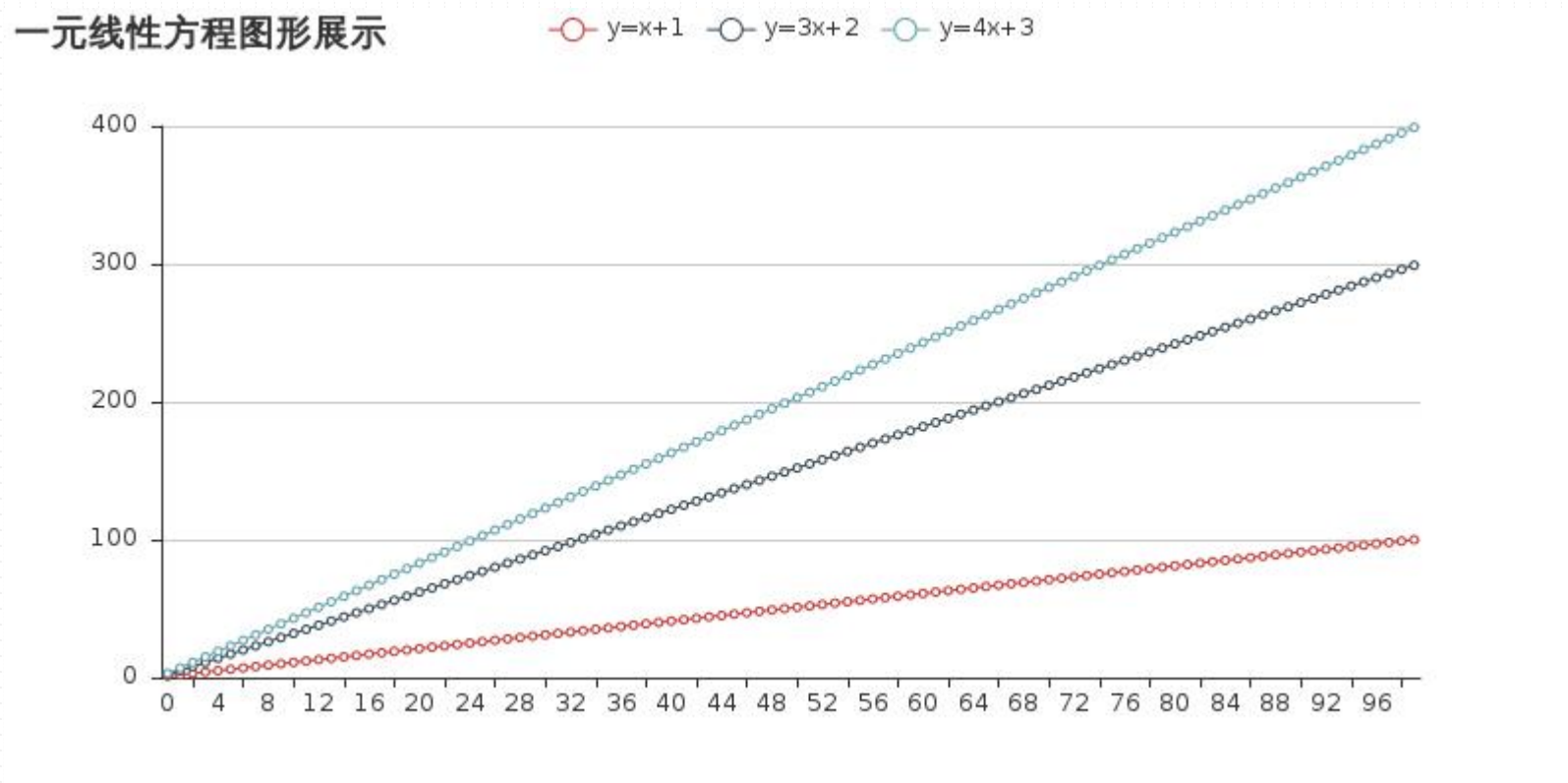
闭包的一个常用场景就是装饰器。



```
def line_conf(a, b):  
    def line(x):  
        return a*x + b  
    return line  
  
line1 = line_conf(1, 1)  
line2 = line_conf(4, 5)  
print(line1(5))  
print(line2(5))
```

函数`line`与变量`a,b`构成闭包。在创建闭包的时候,我们通过`line_conf`的参数`a,b`说明了这两个变量的取值,这样,我们就确定了函数的最终形式($y=x+1$ 和 $y=4x+5$)。

优点: 闭包也具有提高代码可复用性的作用。



装饰器**本质**上是一个函数，该函数用来处理其他函数，它可以让其他函数在不需要修改代码的前提下**增加额外的功能**，装饰器的返回值也是一个函数对象。它经常用于有切面需求的场景，比如：插入日志、性能测试、事务处理、缓存、权限校验等应用场景。



照片与相框的关系

为什么需要装饰器？

写代码要遵循 **开放封闭** 原则,虽然在这个原则是用的面向对象开发,但是也适用于函数式编程,简单来说,它规定已经实现的功能代码不允许被修改,但可以被扩展,即:

- 1.封闭:已实现的功能代码块
- 2.开放:对扩展开发



04

装饰器Decorator

```
userAge = 40

def canYou(func):
    def decorator(*args, **kwargs):
        if userAge > 1 and userAge < 10:
            return func(*args, **kwargs)
        print('你的年龄不符合要求，不能看')
    return decorator

@canYou
def play():
    print('开始播放动画片 《喜洋洋和灰太狼》')

play()
```

1. 引入日志
2. 函数执行时间统计
3. 执行函数前预备处理
4. 执行函数后清理功能
5. 权限校验等场景
6. 缓存

04

装饰器Decorator

装饰器示例: 无参数的函数

04

装饰器Decorator

装饰器示例: 被装饰的函数有参数

04

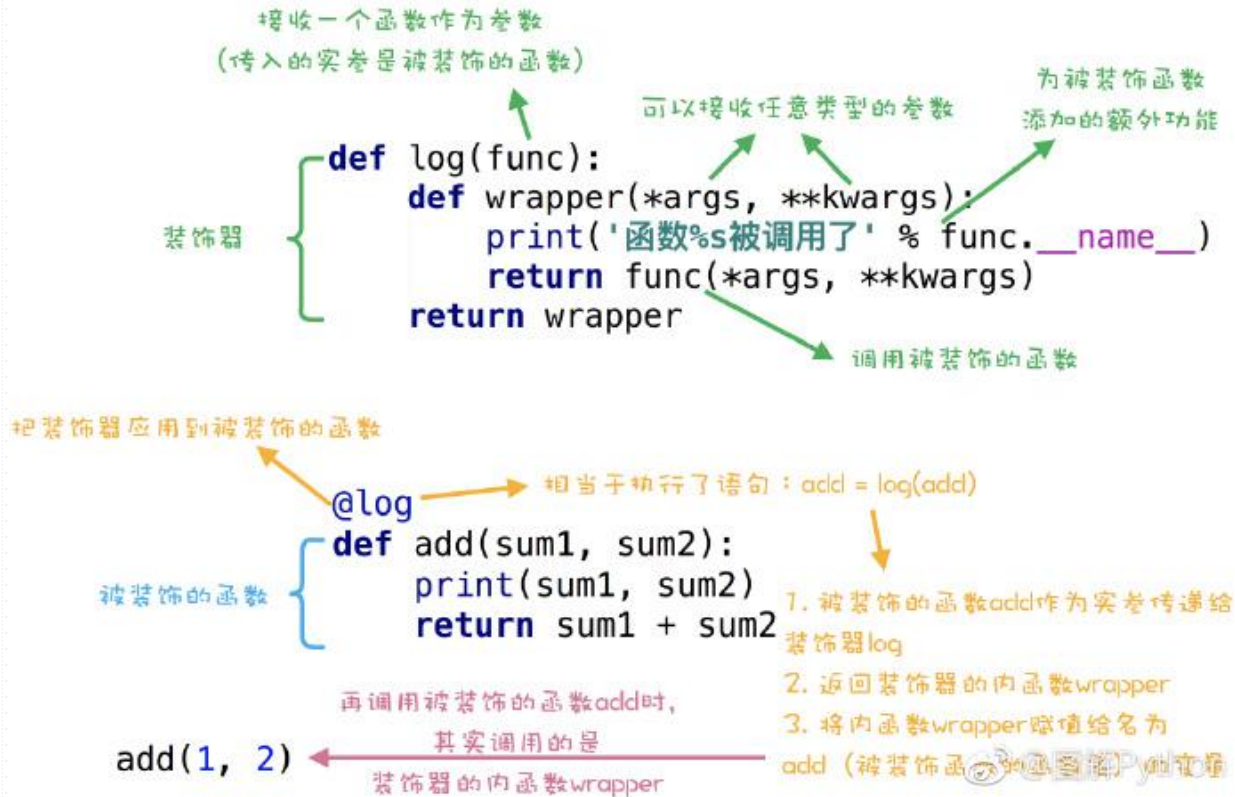
装饰器Decorator

装饰器示例: 被装饰的函数有可变参数和关键字参数

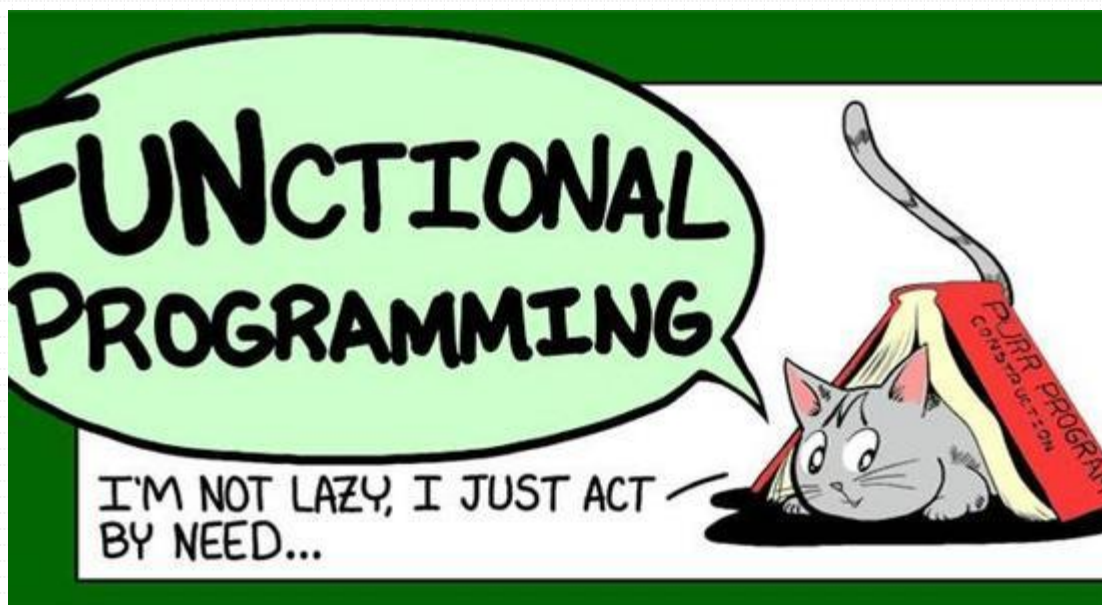
04

装饰器Decorator

装饰器示例: 被多个装饰器装饰的函数



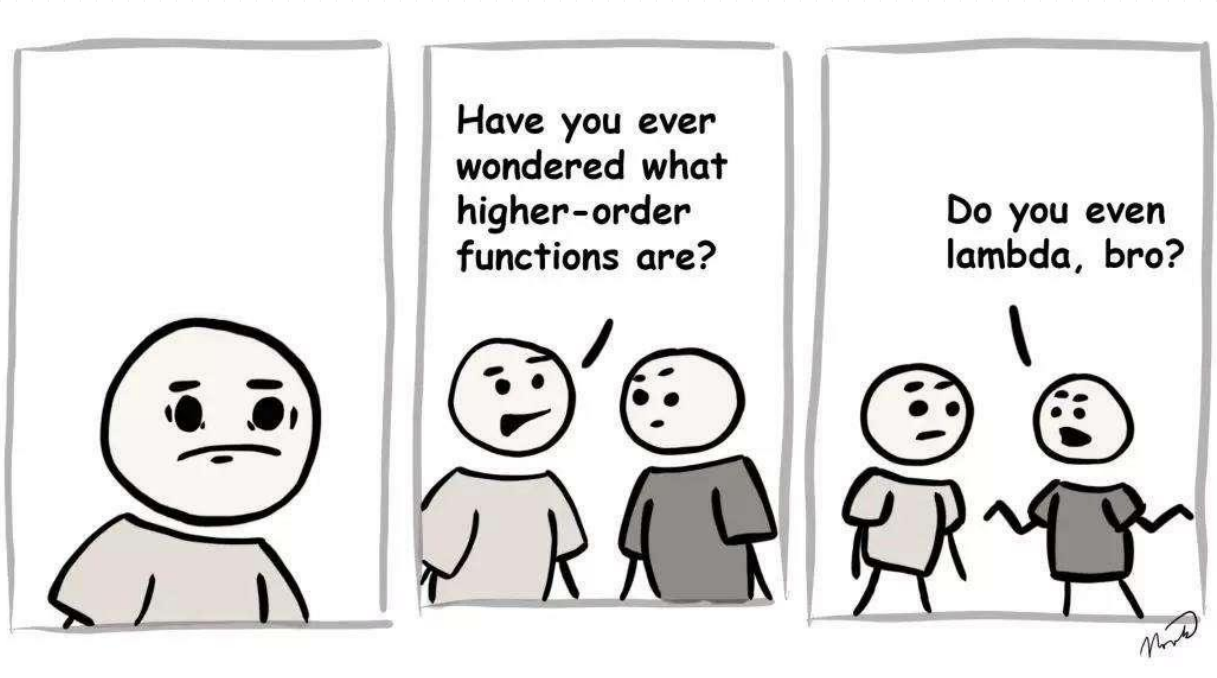
函数式编程的一个特点就是,允许把函数本身作为参数传入另一个函数,还允许返回一个函数!Python对函数式编程提供部分支持。



内置高阶函数

高阶函数英文叫Higher-order function。什么是高阶函数？

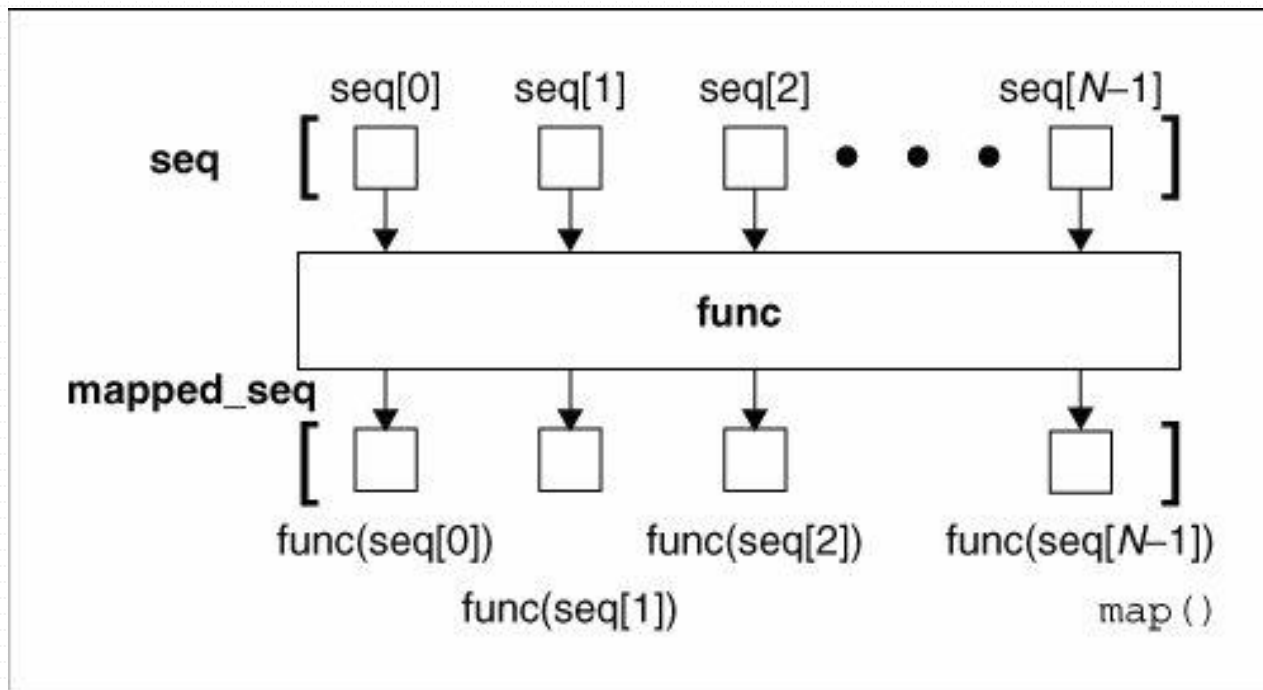
把函数作为参数传入,这样的函数称为高阶函数,函数式编程就是指这种高度抽象的编程范式。



你想过什么是高阶函数吗？

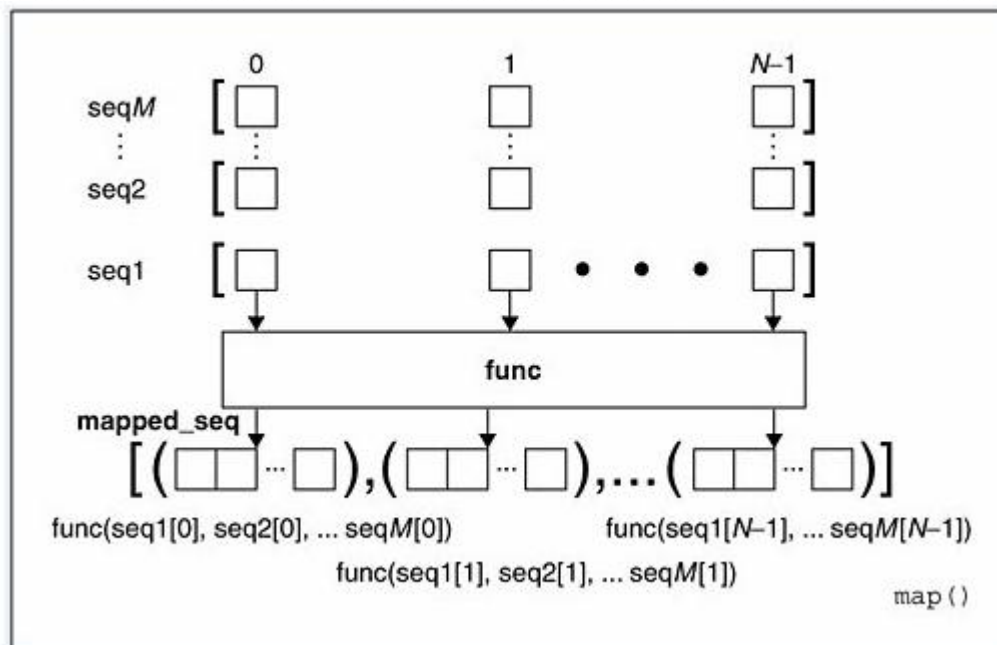
`map()` 会根据提供的函数对指定序列做映射。

第一个参数接受一个函数名，后面的参数接受一个或多个可迭代的序列，返回的是一个集合。



内置高阶函数

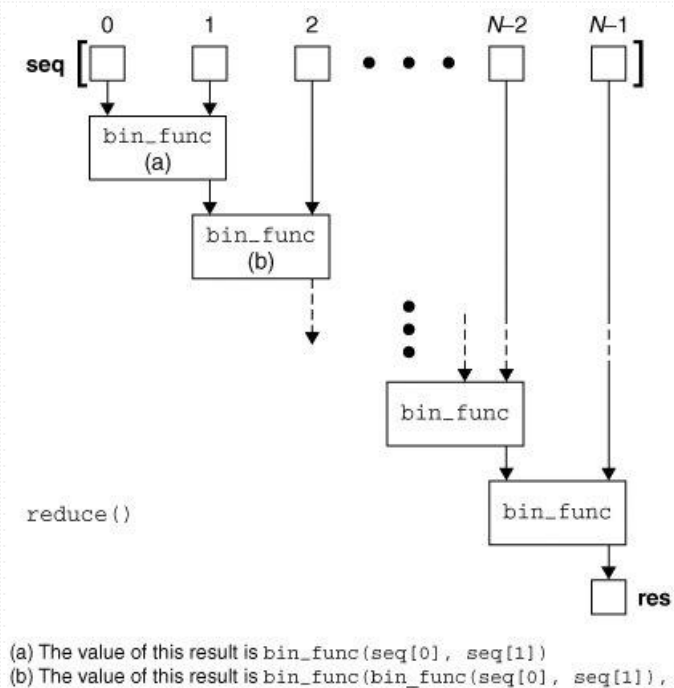
当seq多于一个时，map可以并行（注意是并行）地对每个seq执行如下图所示的过程：



注意: `map`无法处理`seq`长度不一致、对应位置操作数类型不一致的情况，这两种情况都会报类型错误。

`reduce()` 函数会对参数序列中元素进行累积。

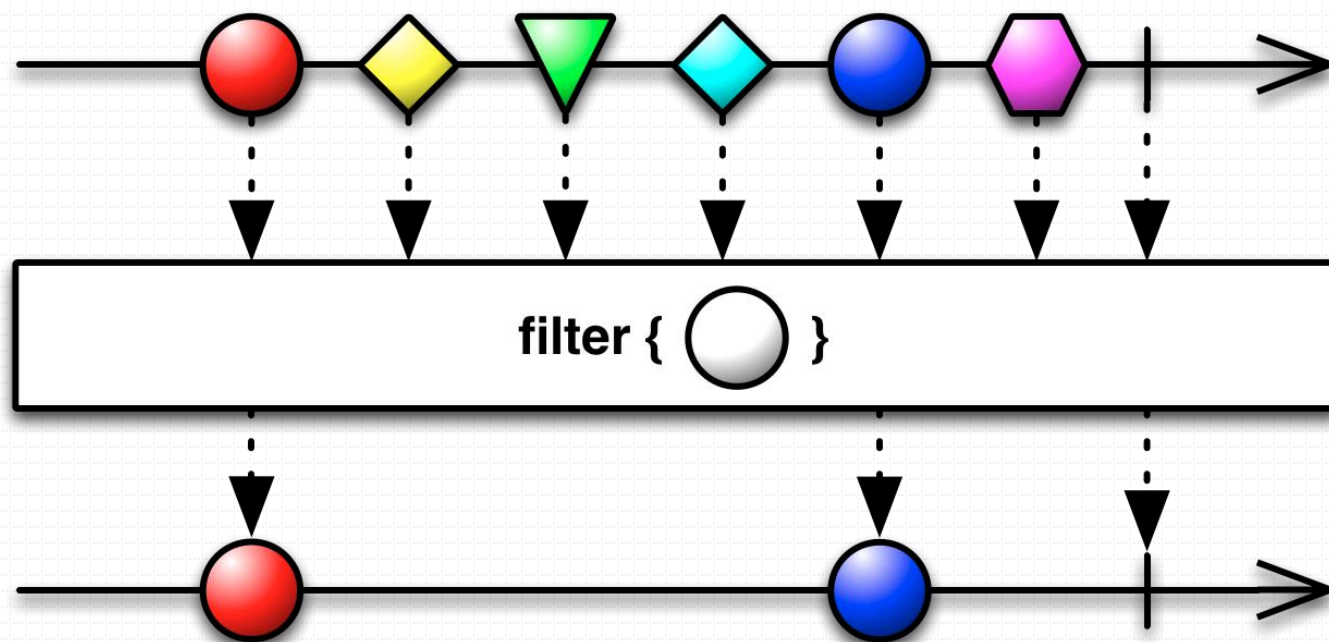
第一个参数接受一个函数名，后面的参数接受一个或多个可迭代的序列，返回的是一个结果。



内置高阶函数

`filter()` 函数用于过滤序列，过滤掉不符合条件的元素，返回由符合条件元素组成的新列表。

第一个参数接受一个函数名，后面的参数接受一个或多个可迭代的序列，返回的是一个序列。



内置高阶函数sorted/max/min

sorted() 函数对所有可迭代的对象进行排序操作。返回重新排序的列表。

sorted(iterable, key=None, reverse=False)

key: 主要是用来进行比较的元素，只有一个参数，

reverse: 排序规则，True 降序，False 升序（默认）。



SIMPLY ORGANISED



python排序`sort()`和`sorted()`的区别是什么？

1. 排序对象不同: `sort` 是应用在 `list` 上的方法, `sorted` 可以对所有可迭代的对象进行排序操作。

2. 返回值不同:

`list` 的 `sort` 方法返回的是对已经存在的列表进行操作, 无返回值,

内建函数 `sorted` 方法返回的是一个新的 `list`, 而不是在原来的基础上进行的操作。

内置高阶函数

1. 对于序列每个元素求绝对值;
2. 接收用户输入3个数字；依次将接收的三个数转换为整形
3. 对于序列每个元素求阶乘; 5!
4. 拿出1~100之间所有的素数

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79
83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163
167 173 179 181 191 193 197 211 223 227 229 233 239 241
251 257 263 269 271 277 281 283 307 311 313 317 331 337
347 349 353 359 367 373 379 383 397 401 409 419 421 431
433 439 443 449 457 461 463 467 479 487 491 499 503 509 521
523 541 547 557 563 569 571 577 587 593 599 601 607 613 617
619 631 641 643 647 653 659 661 673 677 683 691 701 709 719
727 733 739 743 751 757 761 769 773 787 797 809 811 821 823
827 829 839 853 857 859 863 877 881 883 887 907 911 919 929
937 941 947 953 967 971 977 983 991 997 1009 1013 1019 1021
1031 1033 1039 1049 1051 1061 1063 1069 1087 1091 1093
1097 1103 1109 1117 1123 1129 1151 1153 1163 1171 1181
```



感谢聆听！

THANK YOU!