

python 魔鬼训练营 第4周

DATAGURU专业数据分析社区

【声明】 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

<http://edu.dataguru.cn>

第4课：yield表达式

课程内容：

- 特点
- 优势
- 应用

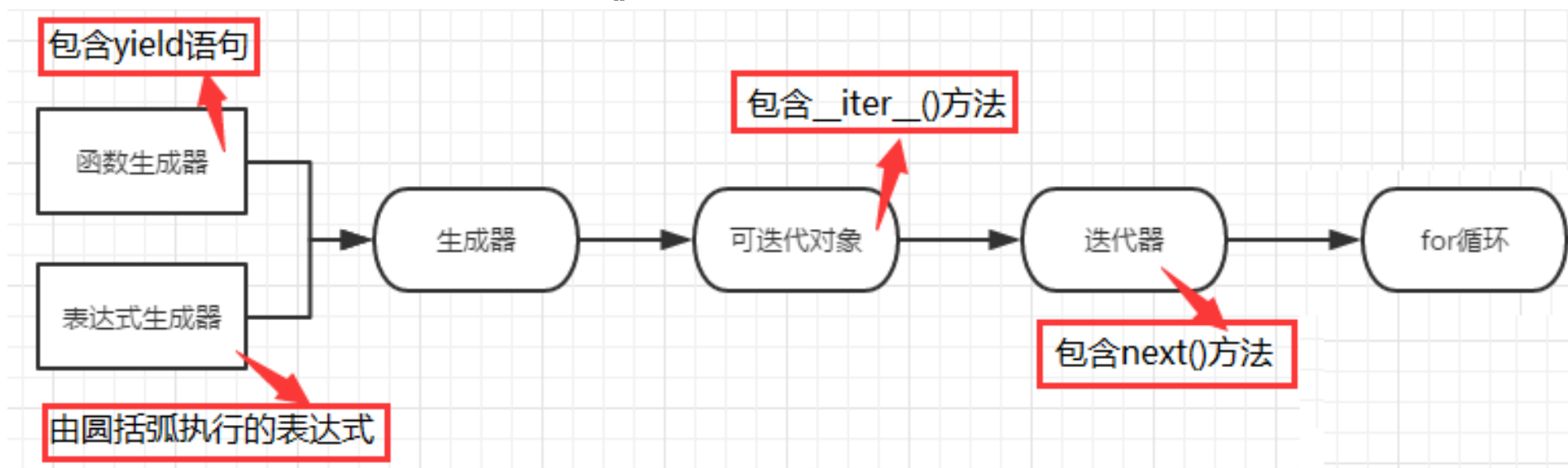
本次课内容重点讲解Python的yield表达式，因为这个关键字平常不常用，也容易产生歧义，不易理解；所以这里把它单独拿出来讲解，它与return的区别，它和生成器的关系，它的通常用法。

第1节：yield特点

- 包含yield的函数是一个生成器
- 返回一个可迭代对象
- 可以接收参数

第1节：生成器、可迭代对象、迭代器

- 生成器：一个可以生成可迭代对象的函数或表达式
- 可迭代对象：一个包含`__iter__`方法的对象
- `__iter__()`：一个返回迭代器的方法
- 迭代器：一个包含`next()`方法的对象



第2节：yield优势

■ return + list

```
def fab(max):
```

```
    l = []
```

```
    n, a, b = 0, 0, 1
```

```
    while n < max:
```

```
        l.append(b)
```

```
        a, b = b, a + b
```

```
        n = n + 1
```

```
    return l
```

第2节：yield优势

■ yield

```
def fab1(max):
```

```
    n, a, b = 0, 0, 1
```

```
    while n < max:
```

```
        yield b
```

```
        a, b = b, a + b
```

```
        n = n + 1
```

第3节：yield应用

■ range与xrange

range → 返回列表

xrange → 返回xrange可迭代对象, iter() → rangeiterator

■ 文件读写

readall → 返回所有行组成的列表

for i in fopen → 返回当前行内容

第3节：yield应用

```
def read_file(fpath):  
    BLOCK_SIZE = 1024  
    with open(fpath, 'rb') as f:  
        while True: block = f.read(BLOCK_SIZE)  
        if block:  
            yield block  
        else:  
            return
```

第3节：yield应用

■ next与send

next: 获取下一个迭代内容

send : 获取下一个迭代内容并发送一个消息过去

next == send(None)

第3节：yield应用

■ 中断生成器

```
def close(self):  
    try:  
        self.throw(GeneratorExit)  
    except (GeneratorExit, StopIteration):  
        pass  
    else:  
        raise RuntimeError("generator ignored GeneratorExit")
```

第4课：yield表达式

要点回顾：

- yield作用
- 与return的区别

- Dataguru (炼数成金) 是专业数据分析网站 , 提供教育 , 媒体 , 内容 , 社区 , 出版 , 数据分析业务等服务。我们的课程采用新兴的互联网教育形式 , 独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围 , 重竞争压力的特点 , 同时又发挥互联网的威力打破时空限制 , 把天南地北志同道合的朋友组织在一起交流学习 , 使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本 , 直线下降至百元范围 , 造福大众。我们的目标是 : 低成本传播高价值知识 , 构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情 , 请看我们的培训网站 <http://edu.dataguru.cn>

Thanks

FAQ时间