

```

1 import random
2 import numpy as np
3 from pathlib import Path
4 from PIL import Image

```

1. 后项需要前项导出，且无法通过列表推导式生成。例如，时间序列中的“随机游走”便是一种满足上述条件的序列数据。其公式为 $X_t = \mu + X_{t-1} + w_t$ ，其中 $\mu$ 为漂移量， $w_t$ 是满足某种条件的独立同分布的随机变量，这里假设其服从正态分布  $N(0, \sigma^2)$ 。本题要求写出实现该功能的迭代器函数。具体要求如下：1. 实现 `random_walk` 生成器，输入参数 $\mu$ ， $X_0$ ， $\sigma^2$ ， $N$ ，函数将迭代返回  $N$  个随机游走生成的变量。2. 利用 `zip`，实现拼合多个 `random_walk` 的生成器，以生成一组时间上对齐的多维随机游走序列。

```

7 def random_walk(mu, x0, sigma2, n):
8     for i in range(n):
9         w = random.gauss(0, sigma2**0.5)
10        x0 = mu + x0 + w
11        yield x0
12
13 def random_walks(mu, sigma2, n, *args):
14     walk = (random_walk(mu, i, sigma2, n) for i in args)
15     return zip(*walk)

```

```

17 for walk in random_walks(0, 1, 5, 1, 2, 3):
18     print(list(walk))

```

```

[0.9181011801424857, 2.5460318206244743, 3.1272791024447675]
[1.3372773088705447, 2.349922816302122, 3.2435164995169257]
[3.0020339699603253, 2.7400129284355916, 2.904383019360891]
[2.2486635157953185, 2.1176395047531837, 3.669221750493002]
[-0.2585991069907818, 2.6453635008260505, 3.2819054123666382]

```

2. 需要迭代的内容数据量过大，无法一次性加载。例如，在图像相关的深度学习任务中，由于数据总量过大，一次性加载全部数据耗时过长、内存占用过大，因此一般会采用批量加载数据的方法。（注：实际应用中由于需要进行采样等操作，通常数据加载类的实现原理更接近字典，例如 pytorch 中的 Dataset 类。）现提供文件 FaceImages.zip（<http://vis-www.cs.umass.edu/fddb/originalPics.tar.gz>，其中包含 5000 余张人脸图片。要求设计 FaceDataset 类，实现图片数据的加载。具体要求：1. 类接收图片路径列表 2. 类支持将一张图片数据以 ndarray 的形式返回（可以利用 PIL 库实现）。3. 实现 \_\_iter\_\_ 方法。4. 实现 \_\_next\_\_ 方法，根据类内的图片路径列表，迭代地加载并以 ndarray 形式返回图片数据。请实现上述生成器和迭代器并进行测试。

```
22 class FaceDataset:
23     def __init__(self, path):
24         self.image_paths = Path(path).glob('**/*.jpg')
25
26     def __iter__(self):
27         return self
28
29     def __next__(self):
30         try:
31             image = Image.open(next(self.image_paths))
32         except StopIteration:
33             raise StopIteration
34         return np.array(image)
35
36     def load(self, bath_size):
37         images = []
38         for i in range(bath_size):
39             try:
40                 images.append(next(self))
41             except StopIteration:
42                 print('图片已全部加载完毕! ')
43                 break
44         return images
```

```
46 datas = FaceDataset('.')
47 n = 0
48 while 1:
49     l = len(datas.load(20))
50     n += l
51     if l < 20:
52         break
53     print(f'共加载{n}张图片')
```

```
图片已全部加载完毕!
共加载28204张图片
```