

کتابخانه `itertools` در پایتون شامل ابزارهایی برای انجام عملیات مختلف روی تکرارکننده‌ها مانند لیست‌ها، تاپل‌ها، و دیگر اشیاء قابل تکرار (iterable) است. این کتابخانه به شما کمک می‌کند تا عملیات پیچیده‌تری مانند ترکیب‌ها، ترتیب‌ها و تولید توالی‌های بی‌پایان را به راحتی انجام دهید.

1. عملیات روی تکرارکننده‌ها

اتصال چندین تکرارکننده با `itertools.chain()`

- `itertools.chain()` برای اتصال چندین تکرارکننده استفاده می‌شود. این تابع به طور عمقی مقادیر تکرارکننده‌ها را به هم متصل کرده و یک تکرارکننده واحد ایجاد می‌کند.

```
import itertools

# اتصال چندین لیست
result = itertools.chain([1, 2, 3], ['a', 'b', 'c'], (10, 20))
print(list(result))
# خروجی: [3, 2, 1, 'a', 'b', 'c', 10, 20]
```

تولید ترکیب‌ها با `itertools.combinations()`

- `itertools.combinations()` ترکیب‌های ممکن از عناصر یک iterable با طول ثابت را تولید می‌کند. ترکیب‌ها ترتیب‌دار نیستند و عنصرها نمی‌توانند تکراری باشند.

```
import itertools

# تولید ترکیب‌های دوگانه از یک لیست
result = itertools.combinations([1, 2, 3], 2)
print(list(result))
# خروجی: [(3, 2), (3, 1), (2, 1)]
```

تولید ترتیب‌ها با `itertools.permutations()`

- `itertools.permutations()` ترتیب‌های مختلف (پرموتیشن‌ها) از عناصر یک iterable با طول ثابت را تولید می‌کند. برخلاف ترکیب‌ها، در پرموتیشن‌ها ترتیب اهمیت دارد.

```
import itertools

# تولید پرموتیشن‌های دوگانه از یک لیست
result = itertools.permutations([1, 2, 3], 2)
print(list(result))
# خروجی: [(2, 3), (1, 3), (3, 2), (1, 2), (3, 1), (2, 1)]
```

تولید ضرب داخلی با `itertools.product()`

- `itertools.product()` ضرب داخلی یا ضرب دکارتی را تولید می‌کند. این تابع تمام ترکیب‌های ممکن از اعضای تکرارکننده‌ها را برمی‌گرداند.

```
import itertools
```

```
# تولید ضرب داخلی (Cartesian Product)
result = itertools.product([1, 2], ['a', 'b'])
print(list(result))
# خروجی: [(1, 'a'), (1, 'b'), (2, 'a'), (2, 'b')]
```

مدیریت تکرارهای بی‌پایان با `itertools.cycle()` و `itertools.count()`

- `itertools.cycle()` یک تکرارکننده بی‌پایان ایجاد می‌کند که از ابتدا شروع به تکرار عناصر موجود در iterable می‌کند.

```
import itertools
```

```
# تکرار بی‌پایان از یک لیست
result = itertools.cycle([1, 2, 3])
for _ in range(10):
    print(next(result))
# خروجی: 1 3 2 1 3 2 1 3 2 1
```

- `itertools.count()` به طور بی‌پایان مقادیر عددی تولید می‌کند که از یک مقدار شروع می‌شود و به صورت متوالی افزایش می‌یابد.

```
import itertools
```

```
# تولید اعداد از 10 به بعد
result = itertools.count(10, 2)
for _ in range(5):
    print(next(result))
# خروجی: 10 12 14 16 18
```

نتیجه‌گیری

کتابخانه `itertools` امکانات بسیار مفیدی برای انجام عملیات پیچیده و کارآمد روی تکرارکننده‌ها فراهم می‌کند. از جمله این امکانات می‌توان به اتصال تکرارکننده‌ها، تولید ترکیب‌ها و پرموتیشن‌ها، تولید ضرب داخلی و کار با تکرارهای بی‌پایان اشاره کرد. این کتابخانه می‌تواند در بسیاری از موارد که نیاز به انجام عملیات ترکیبی و تکراری روی داده‌ها دارید، مفید واقع شود.