

# zip

تابع `zip()` یک تکرارشونده ایجاد می‌کند که عناصر را از هر یک از آیتراپل‌ها گردآوری می‌کند.

این تابع یک تکرارشونده از تاپل‌ها برمی‌گرداند، به طوری که تاپل نام از هر یک از توالی‌ها یا تکرارشونده‌های ورودی حاوی عنصر نام آن است. تکرارشونده هنگامی که تکرارشونده ورودی کوتاه‌ترین آن به پایان برسد متوقف می‌شود. در صورتی که یک آرگومان تکی داشته باشد، تکرارشونده ای از تاپل‌های یک عضوی برمی‌گرداند. در صورتی که بدون آرگومان ورودی داشته باشد، یک تکرارشونده خالی برمی‌گرداند.

تابع `zip()` معادل زیر است:

```
def zip(*iterables):
    # zip('ABCD', 'xy') --> Ax By
    sentinel = object()
    iterators = [iter(it) for it in iterables]
    while iterators:
        result = []
        for it in iterators:
            elem = next(it, sentinel)
            if elem is sentinel:
                return
            result.append(elem)
        yield tuple(result)
```

`zip()` فقط در صورتی باید با آرگومان‌های طول نامساوی استفاده شود که مقدار انتهایی وجود نداشته باشد. نحوه کار آن را در برخی از مثال‌ها ببینیم:

## مثال‌ها

In [2]:

```
1 x = [1,2,3]
2 y = [4,5,6]
3
4 # Zip the lists together
5 list(zip(x,y))
```

Out[2]:

[(1, 4), (2, 5), (3, 6)]

توجه کنید که تاپل‌ها برگردانده می‌شوند. اگر یکی از تکرارشونده‌ها از دیگری بیشتر باشد چه اتفاقی می‌افتد؟

In [3]:

```
1 x = [1,2,3]
2 y = [4,5,6,7,8]
3
4 # Zip the lists together
5 list(zip(x,y))
```

Out[3]:

```
[(1, 4), (2, 5), (3, 6)]
```

توجه کنید که تابع `zip()` توسط طول کوتاه‌ترین تکرارشونده تعریف می‌شود. به طور کلی توصیه می‌شود از `zip()` برای آیتراپل‌های با طول‌های نامساوی استفاده نکنید مگر اینکه مطمئن باشید که فقط نیاز به ترکیبات جزئی از تاپل‌ها دارید.

چه اتفاقی می‌افتد اگر سعی کنیم دیکشنری‌ها را با هم `zip()` کنیم؟

In [4]:

```
1 d1 = {'a':1, 'b':2}
2 d2 = {'c':4, 'd':5}
3
4 list(zip(d1,d2))
```

Out[4]:

```
[('a', 'c'), ('b', 'd')]
```

این مسئله قابل درک است زیرا به سادگی تکرار از دیکشنری‌ها فقط کلیدها را نتیجه می‌دهد. برای ترکیب کلیدها و مقادیر باید متدهایی را فراخوانی کنیم:

In [5]:

```
1 list(zip(d2,d1.values()))
```

Out[5]:

```
[('c', 1), ('d', 2)]
```

In [6]:

```
1 list(zip(d1,d2.values()))
```

Out[6]:

```
[('a', 4), ('b', 5)]
```

عالی! سرانجام از `zip()` برای تعویض کلیدها و مقادیر دو دیکشنری استفاده کنیم:

In [7]:

```
1 def switcharoo(d1,d2):
2     dout = {}
3
4     for d1key, d2val in zip(d1,d2.values()):
5         dout[d1key] = d2val
6
7     return dout
```

In [8]:

```
1 switcharoo(d1,d2)
```

Out[8]:

```
{'a': 4, 'b': 5}
```

عالی! می‌توانید در بسیاری از مواقع از `zip()` برای صرفه‌جویی زمان در ساخت تاپل‌ها استفاده کنید! در حال حاضر باید یک درک خوب از `zip()` و برخی از موارد استفاده ممکن آن را داشته باشید.