

به تابعی که در داخل تابع دیگر تعریف شده باشد، تابع داخلی گفته می شود :

```
def outer():  
    def inner():  
        #Some code
```

در کد بالا تابع `inner()` یک تابع داخلی است. برای اینکه یک تابع داخلی را به وسیله تابع خارجی آن برگشت دهیم باید بعد از کلمه کلیدی `return`، نام تابع داخلی را بدون پرانتز بنویسیم:

```
def outer():  
    def inner():  
        #Some code  
  
    return inner;
```

به مثالی ساده در مورد توابع داخلی توجه کنید :

```
1 def make_adder(x):  
2     def addFive():  
3         return x + 5  
4     return addFive  
5  
6 result = make_adder(10)  
7 print(result())
```

15

قبل از توضیح کد بالا به این نکته توجه کنید که توابع داخلی به متغیرهای تابع خارجی دسترسی دارد. در کد بالا یک تابع داخلی به نام `addFive()` تعریف شده است که وظیفه آن اضافه کردن عدد 5 به مقداری است که با تابع خارجی ارسال می شود. در خط 6 وقتی تابع `make_adder()` را فراخوانی می کنیم و مقدار آن را در داخل یک متغیر به نام `result` می ریزیم، این متغیر در اصل مقدار برگشتی از تابع `make_adder()` یعنی تابع `addFive()` است. در نتیجه برای چاپ نهایی این مقدار برگشتی باید در جلوی نام `result` در خط 7 علامت پرانتز بگذاریم تا به نوعی تابع `addFive()` فراخوانی شود. پس وقتی مقدار 10 را به تابع `make_adder()` ارسال می کنیم، این عدد در داخل تابع `addFive()` با 5 جمع شده و مقدار 15 برگشت داده می شود. توابع داخلی در مبحث Decorator ها کاربرد دارند، که در درس های آینده توضیح می دهیم.