

پارامترها داده‌های خامی هستند که تابع آنها را پردازش می‌کند و سپس اطلاعاتی را که به دنبال آن هستید، در اختیار شما قرار می‌دهد. فرض کنید پارامترها مانند اطلاعاتی هستند که شما به یک کارمند می‌دهید که بر طبق آنها کارش را به پایان برساند. یک تابع می‌تواند هر تعداد پارامتر داشته باشد. هر پارامتر می‌تواند از انواع مختلف داده باشد. در زیر یک تابع با N پارامتر نشان داده شده است :

```
def functionName(param1, param2, ... paramN):  
    code to execute;
```

پارامترها بعد از نام تابع و بین پرانتزها قرار می‌گیرند. بر اساس کاری که تابع انجام می‌دهد می‌توان تعداد پارامترهای زیادی به تابع اضافه کرد. بعد از فراخوانی یک تابع باید آرگومانهای آن را نیز تأمین کنید. آرگومان‌ها مقادیری هستند که به پارامترها اختصاص داده می‌شوند. اجازه بدهید که یک مثال بزنیم :

```
1 def CalculateSum(number1, number2):  
2     return number1 + number2;  
3  
4 num1 = int(input("Enter the first number: "));  
5 num2 = int(input("Enter the second number: "));  
6  
7 print("Sum = {0}".format(CalculateSum(num1, num2)));
```

```
Enter the first number: 10  
Enter the second number: 5  
Sum = 15
```

در برنامه بالا یک تابع به نام CalculateSum() (خطوط 2-1) تعریف شده است و می‌خواهیم مقدار دو عدد را با این تابع جمع کنیم. در بدنه تابع دستور return نتیجه جمع دو عدد را بر می‌گرداند. در خطوط 4 و 5 برنامه از کاربر دو مقدار را درخواست می‌کند و آنها را داخل متغیرها قرار می‌دهد. حال تابع را که آرگومانهای آن را آماده کرده‌ایم فراخوانی می‌کنیم. مقدار num1 به پارامتر اول و مقدار num2 به پارامتر دوم ارسال می‌شود. حال اگر مکان دو مقدار را هنگام ارسال به تابع تغییر دهیم (یعنی مقدار num2 به پارامتر اول و مقدار num1 به پارامتر دوم ارسال شود) هیچ تغییری در نتیجه تابع ندارد چون جمع خاصیت جابه جایی دارد.

فقط به یاد داشته باشید که باید تعداد آرگومانها هنگام فراخوانی تابع دقیقاً با تعداد پارامترها تعریف شده در تابع مطابقت داشته باشد. بعد از ارسال مقادیر 10 و 5 به پارامترها، پارامترها آنها را دریافت می‌کنند. به این نکته نیز توجه کنید که نام پارامترها طبق قرارداد به شیوه کوهان شتری یا camelCasing (حرف اول دومین کلمه بزرگ نوشته می‌شود) نوشته می‌شود. در داخل بدنه تابع (خط 2) دو مقدار با هم جمع می‌شوند و نتیجه به تابع فراخوان (تابعی که تابع CalculateSum() را فراخوانی می‌کند) ارسال می‌شود. در درس آینده از یک متغیر برای ذخیره نتیجه محاسبات استفاده می‌کنیم ولی در اینجا مشاهده می‌کنید که می‌توان به سادگی نتیجه جمع را نشان داد (خط 7).

در خط 7 تابع CalculateSum() را فراخوانی می‌کنیم و دو مقدار صحیح به آن ارسال می‌کنیم. دو عدد صحیح در داخل تابع با هم جمع شده و نتیجه آنها برگردانده می‌شود. مقدار برگشت داده شده از تابع به وسیله تابع print() نمایش داده می‌شود. و نکته آخر اینکه یک تابع را می‌توان به عنوان آرگومان به تابع دیگر ارسال کرد. به کد زیر توجه کنید:

```
1 def functionA(myFunction):  
2     return myFunction()  
3  
4 def functionB():  
5     return "Hello World!";  
6  
7 print("{0}".format(functionA(functionB)));
```

```
Hello World!
```

در کد بالا ما دو تابع تعریف کرده ایم. که تابع اول یعنی functionA قرار است که یک آرگومان از نوع تابع دریافت کند. برای این منظور بعد از تعریف پارامتر در خط 1 در خط 2 و بعد از نام پارامتر حتماً باید علامت های پرانتز را بنویسید. این بدین معنی است که مقدار برگشتی از تابع functionA یک تابع است. در نتیجه هنگامی که در خط 7 ما functionB را به عنوان آرگومان به تابع functionA می‌دهیم، کلمه functionA در خط 2 جایگزین کلمه myFunction شده و در نتیجه این خط همانند فراخوانی تابع functionB عمل می‌کند (یعنی فراخوانی یک تابع همراه با پرانتزهای آن).