

## بخش 3: توابع (Functions) در پایتون

### 1. تعریف توابع

در زبان پایتون، توابع بلوک‌هایی از کد هستند که می‌توانند یک یا چند عملیات را انجام دهند و می‌توانند ورودی‌هایی دریافت کرده و خروجی‌هایی تولید کنند. توابع به شما این امکان را می‌دهند که کدهای خود را سازماندهی کنید، از تکرار جلوگیری کنید و ساختار کد را قابل فهم‌تر کنید.

#### 1.1. نحوه تعریف و استفاده از توابع

برای تعریف یک تابع در پایتون، از کلمه کلیدی `def` استفاده می‌شود، به دنبال آن نام تابع و سپس پارامترهای تابع (در صورت وجود) داخل پرانتز آورده می‌شود. کد داخل تابع باید با استفاده از فاصله گذاری (indentation) مشخص شود. در نهایت، می‌توان تابع را با استفاده از نام تابع فراخوانی کرد.

#### ساختار تعریف یک تابع:

```
def function_name(parameters):  
    # کدهایی که داخل تابع اجرا می‌شوند  
    return value # مقدار برگشتی از تابع
```

- نام تابع است که شما آن را برای فراخوانی تابع استفاده خواهید کرد. `function_name`
- پارامترهایی که تابع می‌تواند دریافت کند. این پارامترها ورودی‌هایی هستند که تابع برای انجام عملیات به آن‌ها نیاز دارد. `parameters`
- مقداری که تابع به عنوان نتیجه عملیات خود بازمی‌گرداند. `return value`

#### 1.2. معرفی ساختار پایه تابع و فراخوانی آن

به عنوان مثال، یک تابع ساده که دو عدد را جمع می‌کند، به صورت زیر است:

```
def add_numbers(a, b):  
    return a + b
```

در این مثال:

- تابع `add_numbers` دو پارامتر به نام‌های `a` و `b` دریافت می‌کند.
  - داخل تابع، جمع این دو عدد انجام شده و نتیجه آن با استفاده از دستور `return` به تابع بازگشت می‌کند.
  - تابع بعد از بازگشت از دستور `return` پایان می‌یابد و مقدار برگشتی را به فراخوانی تابع ارسال می‌کند.
- برای فراخوانی این تابع و استفاده از آن، می‌توان از کد زیر استفاده کرد:

```
result = add_numbers(3, 5)  
print(result)
```

در اینجا:

- تابع `add_numbers(3, 5)` را فراخوانی کرده و به آن مقادیر `3` و `5` را به عنوان ورودی می‌دهد.
- نتیجه‌ی بازگشتی تابع (که 8 است) در متغیر `result` ذخیره می‌شود و سپس با دستور `print` چاپ می‌شود.

## 1.3. توابع بدون مقدار برگشتی

توابع در پایتون می‌توانند مقدار برگشتی نداشته باشند. در چنین حالتی، تابع می‌تواند عملیاتی را انجام دهد، اما چیزی به عنوان نتیجه برنگرداند. به طور پیش‌فرض، در صورتی که تابع هیچ مقدار برگشتی نداشته باشد، پایتون مقدار `None` را باز می‌گرداند.

مثال از تابع بدون مقدار برگشتی:

```
def print_greeting(name):
    print(f"Hello, {name}!")
```

این تابع هیچ مقدار بازگشتی ندارد، بلکه صرفاً یک پیام را چاپ می‌کند. می‌توان این تابع را به شکل زیر فراخوانی کرد:

```
print_greeting("Alice")
```

خروجی برنامه:

```
Hello, Alice!
```

## 1.4. نکات مهم هنگام تعریف توابع

- نام تابع باید یک نام معتبر (که باید با یک حرف یا آندرلاین شروع شود) باشد و نباید نام‌های کلیدی پایتون (مثل `def`, `return`, `True` و ...) را استفاده کرد.
- پارامترها می‌توانند به عنوان ورودی برای انجام محاسبات یا عملیات‌های دیگر در داخل تابع استفاده شوند.
- اگر در داخل تابع از دستور `return` استفاده شود، تابع بلافاصله پایان می‌یابد و نتیجه‌ای باز می‌گرداند. در غیر این صورت، تابع به صورت پیش‌فرض `None` را باز می‌گرداند.

## 1.5. فراخوانی توابع

پس از تعریف یک تابع، برای استفاده از آن، فقط کافی است نام تابع را همراه با پارامترهای مورد نظر (در صورت نیاز) فراخوانی کنیم. توابع می‌توانند در هر جایی از برنامه (پس از تعریف شدن) فراخوانی شوند.

```
def multiply(a, b):
    return a * b

result = multiply(4, 6)
print(result)
```

در اینجا:

- تابع `multiply` تعریف شده است که دو عدد را به هم ضرب می‌کند.
- سپس این تابع فراخوانی می‌شود و نتیجه‌ی آن (24) در متغیر `result` ذخیره می‌شود و چاپ می‌شود.

خروجی برنامه:

در این بخش، شما یاد گرفتید که چگونه توابع را در پایتون تعریف کنید، چطور آن‌ها را فراخوانی کنید و مقادیر بازگشتی را از آن‌ها دریافت کنید. این مبانی به شما کمک خواهد کرد تا برنامه‌های پیچیده‌تری بسازید و کدهای خود را سازماندهی کنید.

## پارامترها و مقادیر بازگشتی (Return)

در این بخش به نحوه تعریف پارامترهای تابع و نحوه استفاده از آن‌ها برای دریافت داده‌ها و همچنین نحوه بازگشت مقادیر از تابع با استفاده از دستور `return` پرداخته خواهد شد. همچنین در مورد بازگشت بیش از یک مقدار از تابع نیز توضیح داده می‌شود.

### 1. تعریف پارامترهای تابع و نحوه استفاده از آن‌ها

پارامترها، ورودی‌هایی هستند که به تابع داده می‌شوند تا در هنگام اجرای تابع از آن‌ها استفاده شود. هر تابع می‌تواند یک یا چند پارامتر داشته باشد که هنگام فراخوانی تابع، مقادیر مشخصی به آن‌ها ارسال می‌شود.

ساختار پارامترهای تابع:

```
def function_name(parameter1, parameter2):  
    # عملیات روی پارامترها  
    return result
```

در اینجا:

- `parameter1` و `parameter2` پارامترهای تابع هستند.
- هنگام فراخوانی تابع، مقادیر خاصی به این پارامترها ارسال می‌شود.

مثال:

```
def greet(name, age):  
    print(f"Hello {name}, you are {age} years old.")  
  
greet("Alice", 30)
```

در اینجا:

- تابع `greet` دو پارامتر به نام‌های `name` و `age` دارد.
- هنگامی که تابع فراخوانی می‌شود، مقادیر `"Alice"` و `30` به ترتیب به پارامترهای `name` و `age` ارسال می‌شود.

خروجی برنامه:

```
Hello Alice, you are 30 years old.
```

## 2. نحوه بازگشت مقادیر از تابع با استفاده از `return`

دستور `return` برای بازگشت مقدار از یک تابع استفاده می‌شود. هنگامی که یک تابع به دستور `return` می‌رسد، مقدار بازگشتی به فراخوانی تابع باز می‌گردد و اجرای تابع پایان می‌یابد. شما می‌توانید با استفاده از `return` هر نوع داده‌ای از جمله اعداد، رشته‌ها، لیست‌ها و غیره را بازگردانید.

ساختار دستور `return`:

```
def function_name():  
    return value
```

مثال:

```
def add(a, b):  
    return a + b  
  
result = add(3, 5)  
print(result)
```

در اینجا:

- تابع `add` دو پارامتر `a` و `b` را دریافت می‌کند و جمع آن‌ها را بازمی‌گرداند.
- دستور `return a + b` نتیجه جمع را بازمی‌گرداند.
- مقدار بازگشتی به متغیر `result` اختصاص داده می‌شود و سپس چاپ می‌شود.

خروجی برنامه:

8

## 3. کاربرد `return` برای بازگشت بیش از یک مقدار (از طریق `Tuple`)

پایتون این امکان را فراهم می‌کند که یک تابع بیش از یک مقدار را از طریق یک ساختار داده ترکیبی مثل `tuple` بازگرداند. شما می‌توانید چندین مقدار را در یک `tuple` قرار داده و آن را به عنوان خروجی از تابع بازگردانید.

ساختار بازگشت چندین مقدار:

```
def function_name():  
    return value1, value2, value3
```

در اینجا:

- چندین مقدار از تابع بازگشت داده می‌شود، و پایتون به طور خودکار آن‌ها را در یک `tuple` قرار می‌دهد.

مثال:

```
def calculate(a, b):
    sum_result = a + b
    difference = a - b
    product = a * b
    return sum_result, difference, product

result = calculate(10, 5)
print(result)
```

در اینجا:

- تابع `calculate` سه مقدار را محاسبه کرده و آن‌ها را به صورت یک `tuple` باز می‌گرداند.
- نتیجه به متغیر `result` اختصاص داده می‌شود و سپس چاپ می‌شود.

خروجی برنامه:

```
(15, 5, 50)
```

در صورتی که بخواهید از مقادیر بازگشتی به صورت جداگانه استفاده کنید، می‌توانید از بازگشت چندگانه (Multiple Assignment) استفاده کنید:

```
sum_result, difference, product = calculate(10, 5)
print(sum_result, difference, product)
```

خروجی برنامه:

```
15 5 50
```

نکات:

- در پایتون، هنگامی که چند مقدار را با استفاده از `return` باز می‌گردانید، این مقادیر به طور خودکار در یک `tuple` قرار می‌گیرند.
- شما می‌توانید مقادیر بازگشتی را به صورت مجزا نیز به متغیرهای مختلف اختصاص دهید.

در این بخش، شما یاد گرفتید که چگونه پارامترها را به توابع ارسال کرده و از آن‌ها استفاده کنید و همچنین چطور می‌توانید از دستور `return` برای بازگشت مقادیر از تابع استفاده کنید. همچنین با نحوه بازگشت چندین مقدار با استفاده از `tuple` آشنا شدید که می‌تواند به شما کمک کند تا داده‌های مختلف را به راحتی از توابع بازگردانید.

## توابع بدون مقدار برگشتی

توابع بدون مقدار برگشتی، توابعی هستند که تنها یک یا چند عملیات خاص را انجام می‌دهند و نیازی به بازگشت داده ندارند. این توابع می‌توانند عملیات‌هایی مانند چاپ اطلاعات، تغییر وضعیت متغیرهای خارجی، یا اجرای سایر وظایف جانبی را انجام دهند.

## 1. تعریف توابع بدون مقدار برگشتی

یک تابع بدون مقدار برگشتی، معمولاً با استفاده از دستور `return` به هیچ مقداری باز نمی‌گردد. در این حالت، تابع فقط عملیات خاصی را انجام می‌دهد و سپس به پایان می‌رسد. اگر تابع هیچ مقداری را برنگرداند، به صورت پیش‌فرض مقدار `None` باز می‌گردد، ولی این مقدار معمولاً در توابع بدون مقدار برگشتی مورد استفاده قرار نمی‌گیرد.

ساختار تابع بدون مقدار برگشتی:

```
def function_name():  
    # انجام عملیات  
    print("This is a function with no return value.")
```

در اینجا:

- تابع `function_name` هیچ مقدار برگشتی ندارد.
- تنها عملیات داخل آن انجام می‌شود.

مثال:

```
def greet(name):  
    print(f"Hello, {name}!")  
  
greet("Alice")
```

در اینجا:

- تابع `greet` تنها عملیاتی مانند چاپ پیامی به کنسول انجام می‌دهد و هیچ مقدار بازگشتی ندارد.

خروجی برنامه:

```
Hello, Alice!
```

## 2. استفاده از `print` در توابع بدون مقدار برگشتی

تابع بدون مقدار برگشتی می‌تواند برای نمایش اطلاعات به کاربر از دستور `print` استفاده کند. این کار به خصوص برای توابعی که تنها به خروجی اطلاعات نیاز دارند و نیازی به بازگشت مقدار ندارند، مفید است.

مثال:

```
def display_sum(a, b):  
    result = a + b  
    print(f"The sum of {a} and {b} is {result}")  
    # Write File python for example  
  
display_sum(10, 5)
```

در اینجا:

- تابع `display_sum` دو عدد را دریافت کرده و حاصل جمع آن‌ها را چاپ می‌کند.
- این تابع هیچ مقدار بازگشتی ندارد، بلکه فقط نتیجه عملیات را به صورت خروجی در کنسول نمایش می‌دهد.

خروجی برنامه:

The sum of 10 and 5 is 15

### 3. اعمال تغییرات در متغیرهای خارجی در توابع بدون مقدار برگشتی

در پایتون، می‌توان در توابع بدون مقدار برگشتی، تغییراتی را در متغیرهای خارجی (متغیرهایی که خارج از تابع تعریف شده‌اند) اعمال کرد. این عملیات معمولاً برای تغییر وضعیت داده‌ها یا انجام محاسبات در خارج از تابع کاربرد دارد.

مثال:

```
def update_value(a):  
    a += 5 # تغییر مقدار متغیر a  
    print(f"Updated value: {a}")  
  
x = 10  
update_value(x)  
print(f"Value of x after function call: {x}")
```

در اینجا:

- تابع `update_value` مقدار ورودی `a` را تغییر می‌دهد و پس از تغییر آن را چاپ می‌کند.
- با وجود این‌که متغیر `x` به تابع ارسال می‌شود، این تغییرات تنها در داخل تابع اعمال می‌شود، زیرا متغیر `x` به عنوان یک مقدار (By Value) به تابع ارسال شده است.

خروجی برنامه:

```
Updated value: 15  
Value of x after function call: 10
```

در اینجا:

- مقدار `x` در داخل تابع تغییر می‌کند، ولی چون در پایتون متغیرها به صورت مقادیر ارسال می‌شوند (نه به صورت ارجاع)، تغییرات فقط در داخل تابع مؤثر هستند.
  - برای تغییر متغیرهای خارجی از نوع "ارجاعی" (مثل لیست‌ها یا دیکشنری‌ها) در توابع بدون مقدار برگشتی، می‌توان مستقیماً آن‌ها را تغییر داد:
- مثال با لیست‌ها (متغیر ارجاعی):

```
def append_to_list(lst, item):  
    lst.append(item)  
  
my_list = [1, 2, 3]  
append_to_list(my_list, 4)  
print(my_list)
```

در اینجا:

- تابع `append_to_list` به لیست `lst` یک آیتم جدید اضافه می‌کند.
- چون لیست‌ها به صورت ارجاعی (By Reference) به توابع ارسال می‌شوند، تغییرات در داخل تابع مستقیماً در متغیر خارجی `my_list` اعمال می‌شود.

خروجی برنامه:

```
[1, 2, 3, 4]
```

## نکات:

- توابع بدون مقدار برگشتی معمولاً برای انجام وظایف جانبی مانند چاپ اطلاعات یا تغییر وضعیت داده‌ها استفاده می‌شوند.
- دستور `return` در توابع بدون مقدار برگشتی معمولاً وجود ندارد، اما اگر هم وجود داشته باشد، به طور پیش‌فرض مقدار `None` بازمی‌گرداند.
- در پایتون، متغیرها به صورت مقادیر به توابع ارسال می‌شوند (برای متغیرهای عددی، رشته‌ها و تاپل‌ها)، اما برای انواع داده ارجاعی مانند لیست‌ها و دیکشنری‌ها، تغییرات به صورت مستقیم در خارج از تابع اعمال می‌شود.

---

در این بخش، شما یاد گرفتید که توابع بدون مقدار برگشتی چگونه عمل می‌کنند و در چه مواقعی از آن‌ها استفاده می‌شود. همچنین به شما نشان داده شد که چگونه می‌توانید از دستور `print` برای نمایش خروجی‌ها و از متغیرهای خارجی برای ذخیره تغییرات در توابع بدون بازگشت استفاده کنید.