Functions

یک **تابع** بخشی از کد است که تنها در صورتی اجرا می‌شود که آن را فراخوانی کنید. توابع یکی از مفاهیم اصلی برنامه‌نویسی هستند که امکان سازماندهی بهتر و استفاده مجدد از کد را فراهم می‌کنند.

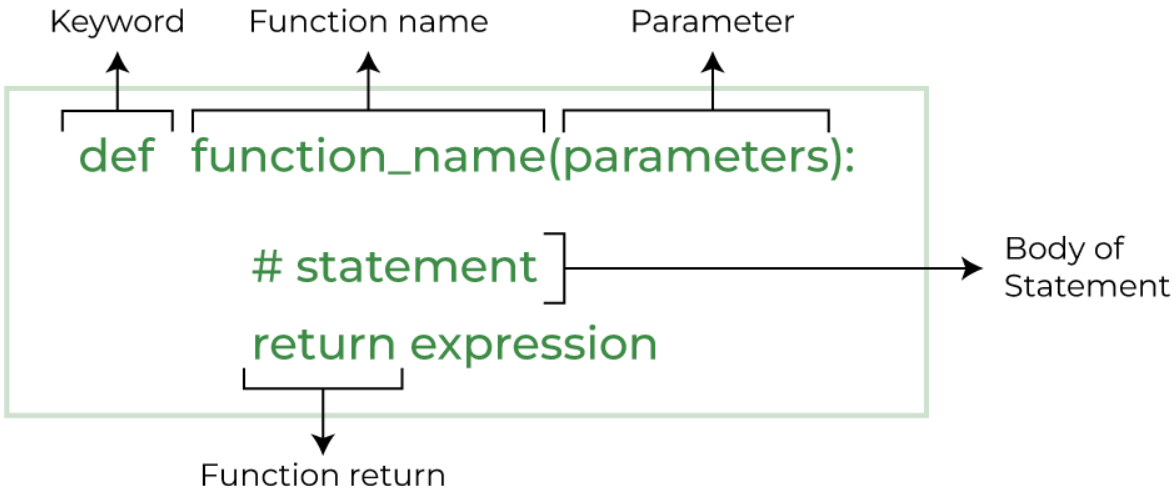
ویژگی‌های توابع:

۱. می‌توانید داده‌هایی را که به آن‌ها «پارامتر» گفته می‌شود، به یک تابع ارسال کنید.

۲. یک تابع می‌تواند پس از اجرا، داده‌ای را به‌عنوان **نتیجه** بازگرداند.

۳. توابع به شما این امکان را می‌دهند که کدهای تکراری را کاهش داده و برنامه‌های خود را مرتب‌تر و قابل نگهداری‌تر کنید.

برای نوشتن یک تابع در پایتون، کافی است دستور ساخت آن را بدانید. این دستور به صورت زیر است:



مقادیر بازگشتی

برای اینکه یک تابع مقداری را بازگرداند، از دستور `return` استفاده کنید:

```
1  def my_function(x):
2      return 5
3
4
```

```
print(my_function(5))
print(my_function(9))
```

تا اینجا دریافتیم که توابع در پایتون تفاوت چندانی با مفاهیمی که در مبانی کامپیوتر یاد گرفته‌ایم ندارند. در پایتون می‌توان توابع را به‌صورت صریح تعریف کرد:

```
# Define a function that checks if a number is even
def is_even(number):
    # Return True if the number is even, otherwise False
    return number % 2 == 0

# Call the function with an example
num = 4
if is_even(num):
    print(f"{num} is even.")
else:
    print(f"{num} is odd.")
```

همچنین با استفاده از **نوع‌دهی (Type Hinting)** می‌توان ساختار توابع پایتونی را شبیه به توابع زبان C کرد. این ویژگی نه‌تنها خوانایی کد را افزایش می‌دهد، بلکه کدنویسی دقیق‌تر و قابل‌فهم‌تری را فراهم می‌کند. در توابع با استفاده از `:` می‌توان نوع پارامترها و با `->` نوع خروجی تابع را مشخص کرد:

```
# Function like C
def add_numbers(a: int, b: int) -> int:  # Equivalent to: int add_numbers
    return a + b

# Call the function
result = add_numbers(5, 7)  # Equivalent to: int result = add_numbers(5,
print("Result =", result)
```

تعداد آرگومان‌ها (Arguments)

به‌صورت پیش‌فرض، هنگام فراخوانی یک تابع، باید تعداد آرگومان‌های ارسال‌شده با تعداد آرگومان‌های تعریف‌شده در تابع مطابقت داشته باشد. این به این معناست که اگر تابعی انتظار دریافت ۲ آرگومان را داشته

باشد، تنها در صورتی درست عمل می‌کند که دقیقاً با ۲ آرگومان فراخوانی شود؛ نه کمتر و نه بیشتر.

```
def my_function(fname, lname):
    print(fname + " " + lname)
my_function("Homayoun", "Shajarian")
```