Analytics Vidhya

# Functions in Python


Instructor

Prashant Sahu
Manager (DataScience), Analytics Vidhya

# Importance of Function in Python

# Function: Scenario

Anna wants to send a personalized greeting message to a list of 50 people, ensuring every message is consistent except for the recipient's name

```
print('Hello, John! How are you today?')

print('Sending message to Sarah...')
print('Hello, Sarah! How are you today?')

print('Sending message to Mike...')
print('Hello, Mike! How are you today?')

print('Sending message to Lisa...')
```

Simple Approach

```
# Using a function (more efficient)
def send_message(name):
    print(f'Sending message to {name}...')
    print(f'Hello, {name}! How are you today?')

# Calling the function for each person
send_message('John')
send_message('Sarah')
send_message('Mike')
send_message('Lisa')
```

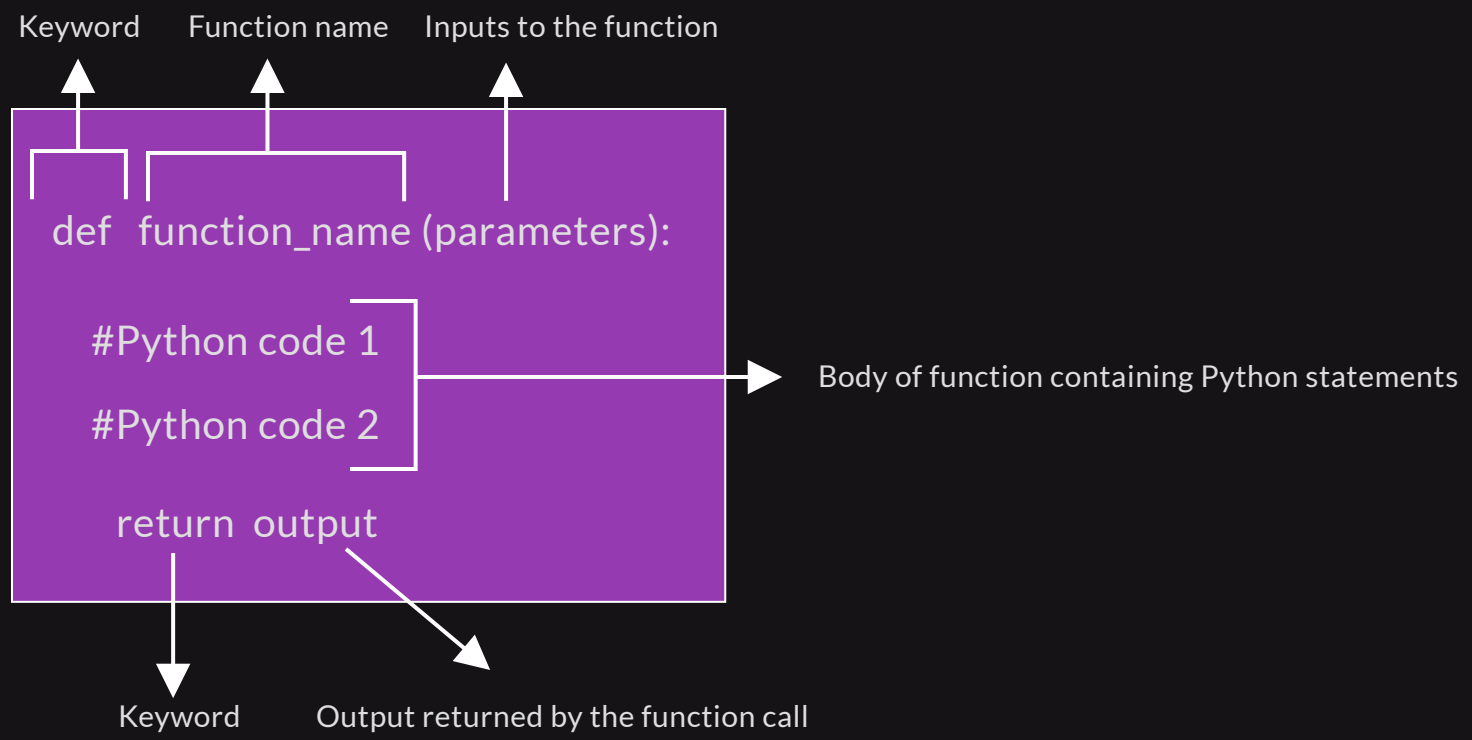
Efficient Approach

Analytics Vidhya

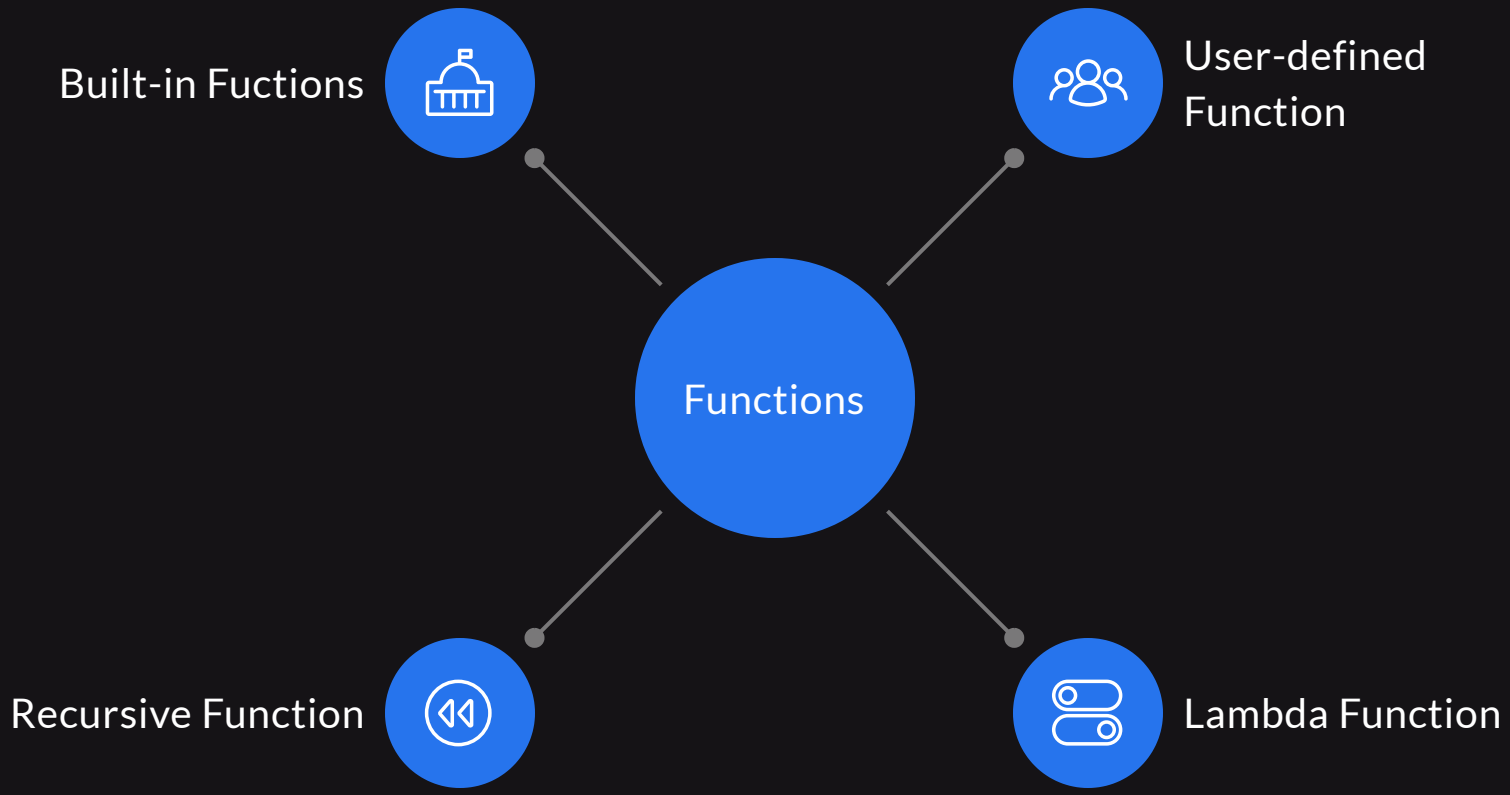
# Function: Introduction

Functions in Pythons is a block of statements that returns the specific tasks.

- Idea is to put some commonly done tasks together and make a function to reuse code contained in it over and over again.




Analytics Vidhya




# Types of Functions

There are various types of functions used to perform different operations.

Analytics Vidhya

# Built-in Function

Python offers a variety of built-in functions that can be used without any imports.

| Input or Output | Datatype conversion | Mathematical Functions | Other Functions |
|---|---|---|---|
| input() | bool(), int(), dict(), float() | abs(), sum(), max(), min() | len() |
| print() | list(), set(), str(), tuple() | pow(), round(), divmod() | range(), type() |

To know more about this follow this link

# User-Defined Function

A user-defined function is created by user to perform specific tasks, rather than using the built-in functions provided by the Python.

To create a function, one need to use the "**def**" keyword followed by the function name and input parameters within parentheses.

```
[1]: def greet(name):
         """This function takes a name and prints a greeting."""
         print(f"Hello, {name}!")

     # Calling the function
     greet("Mayank")

     Hello, Mayank!
```

# Lambda Function

Lambda functions in Python are small, anonymous, and single-use functions that are defined using the "lambda" keyword instead of "def".

- Lambda functions are limited to a single expression, and their results are returned automatically.
- Lambda functions are commonly used in conjuction with higher-order function like map(), filter(), or sorted()

```
# Using a lambda function directly with map()
numbers = [1, 2, 3, 4]
squared = list(map(lambda x: x ** 2, numbers))
print(squared)  # Output: [1, 4, 9, 16]
```

# Recursive Function

A Recursive function calls itself to solve a smaller instance of the same problem.

In Python, recursive functions are typically used when a problem can be defined in terms of itself, like calculating factorials, or generating Fibonacci sequences.

```python
def factorial(n):
    # Base case: if n is 0, return 1
    if n == 0:
        return 1
    # Recursive case: n * factorial(n-1)
    else:
        return n * factorial(n - 1)

# Example usage
print(factorial(5))  # Output: 120
```

# Thank You

Analytics Vidhya