Scope of Function in Python

Local Scope and Global Scope

1. Introduction to Scope

In Python, the scope of a variable refers to the region of the program where that variable can be accessed or used. Understanding scope is essential because it helps avoid errors, improves readability, and ensures that data is managed correctly within a program. The two most commonly used scopes in Python are Local Scope and Global Scope.

2. Local Scope

2.1 Explanation

A local scope refers to variables that are defined inside a function. These variables can only be accessed within that function. Once the function execution is completed, the local variables are destroyed and cannot be accessed outside the function. Local variables help in data security and prevent accidental modification of values.

2.2 Key Characteristics

- Defined inside a function
- Accessible only within that function
- Exists temporarily during function execution
- Same variable name can be reused in different functions

2.3 Examples of Local Scope

Example 1: Student Marks

```
def marks():
    score = 85
    print(score)

marks()
```

Example 2: Temperature Reading

```python
def temperature():
    temp = 30
    print(temp)

temperature()
```

Example 3: Login Attempts

```python
def login():
    attempts = 3
    print(attempts)

login()
```

Example 4: Shopping Bill

```python
def bill():
    total = 1500
    print(total)

bill()
```

Example 5: Exam Duration

```python
def exam_time():
    minutes = 90
    print(minutes)

exam_time()
```

Example 6: Fuel Used

```python
def drive():
    fuel_used = 5
    print(fuel_used)

drive()
```

Example 7: Attendance Count

```python
def attendance():
    present = 27
    print(present)

attendance()
```

Example 8: Password Length

```
def password_check():
    length = 8
    print(length)

password_check()
```

Example 9: Order Quantity

```
def order():
    quantity = 4
    print(quantity)

order()
```

Example 10: Ticket Price

```
def ticket():
    price = 500
    print(price)

ticket()
```

Example 11: Daily Steps

```
def steps():
    count = 6000
    print(count)

steps()
```

Example 12: Battery Usage

```
def phone_use():
    used = 12
    print(used)

phone_use()
```

Example 13: Quiz Score

```
def quiz():
    score = 9
    print(score)

quiz()
```

Example 14: Internet Data

```
def data_used():
    mb = 250
    print(mb)

data_used()
```

Example 15: Delivery Time

```
def delivery():
    hours = 2
    print(hours)

delivery()
```

3. Global Scope

3.1 Explanation

A global scope refers to variables that are defined outside all functions. These variables can be accessed anywhere in the program. However, if a global variable needs to be modified inside a function, the `global` keyword must be used. Global variables remain in memory throughout the program execution.

3.2 Key Characteristics

- Defined outside functions
- Accessible throughout the program
- Exists as long as the program runs
- Requires `global` keyword for modification inside a function

3.3 Examples of Global Scope

Example 1: Bank Balance

```
balance = 1000

def check_balance():
    print(balance)

check_balance()
```

Example 2: Game Score

```
score = 0

def hit():
    global score
    score += 10

hit()
print(score)
```

Example 3: User Role

```
role = "admin"

def access():
    print(role)

access()
```

Example 4: ATM Withdrawal

```
cash = 5000

def withdraw():
    global cash
    cash -= 1000

withdraw()
print(cash)
```

Example 5: App Theme

```
theme = "dark"

def show_theme():
    print(theme)

show_theme()
```

Example 6: Order Status

```
status = "Pending"

def update_status():
    global status
    status = "Delivered"

update_status()
print(status)
```

Example 7: School Name

```
school = "City School"

def info():
    print(school)

info()
```

Example 8: Fuel Tank

```
fuel = 40

def drive():
    global fuel
    fuel -= 5

drive()
print(fuel)
```

Example 9: Cart Items

```
items = 3

def add_item():
    global items
    items += 1

add_item()
print(items)
```

Example 10: Internet Speed

```
speed = 20

def check_speed():
    print(speed)

check_speed()
```

Example 11: Employee Count

```
employees = 15

def show_employees():
    print(employees)

show_employees()
```

Example 12: Website Visitors

```
visitors = 100

def new_visitor():
    global visitors
    visitors += 1

new_visitor()
print(visitors)
```

Example 13: System Mode

```
mode = "online"

def system_status():
    print(mode)

system_status()
```

Example 14: Music Volume

```
volume = 50

def increase_volume():
    global volume
    volume += 10

increase_volume()
print(volume)
```

Example 15: Library Books

```
books = 200

def issue_book():
    global books
    books -= 1
```

```

issue_book()
print(books)
```

4. Conclusion

Local scope variables improve safety and clarity by limiting access to data within functions, whereas global scope variables allow shared access across the program. Proper use of both scopes leads to clean, efficient, and error-free Python programs.