



Lecture slides - Week 6

Programming in Python - Functions and Modules

Dr. Aamir Akbar

Director of both [AWKUM AI Lab](#) and [AWKUM Robotics](#), [Final Year Projects \(FYPs\)](#) coordinator,
and lecturer at the department of Computer Science
Abdul Wali Khan University, Mardan (AWKUM)

1. Introduction
2. Variables Scopes
3. Python Modules
4. Problem Solving Approach with Examples

Introduction

Variables Types and Ids i

In Python, variables have different types, and each variable has a unique identifier (ID) associated with it.

Example 1:

```
x = 42
print(type(x))    # Output: <class 'int'>
print(id(x))      # Output: (varies based on system)
```

Example 2:

```
name = "Alice"
print(type(name)) # Output: <class 'str'>
print(id(name))   # Output: (varies based on system)
```

Example 3:

```
numbers = [1, 2, 3]
print(type(numbers)) # Output: <class 'list'>
print(id(numbers))   # Output: (varies based on system)
```

Remember that the ID of a variable may change if the variable's value changes or if it's reassigned.

Python Functions i

In Python, functions are reusable blocks of code that perform a specific task. They are defined using the `def` keyword and can take zero or more parameters.

Creating a Python Function:

1. Use the `def` keyword followed by the function name.
2. Provide a list of parameters (if any) within parentheses.
3. End the function definition line with a colon `:` to indicate the start of the function body.
4. Indent the function body to specify the code that belongs to the function.
5. Optionally, include a `return` statement to return a value from the function (if needed).

Python Functions ii

Example: Define a function without parameters

```
1 # Define a simple function without parameters
2 def greet():
3     print("Hello, World!")
4
5 # Call the function
6 greet()
```

Example: Define a Function with Parameters:

```
1 # Define a function with parameters
2 def add_numbers(a, b):
3     result = a + b
4     return result
5
6 # Call the function with arguments
7 sum_result = add_numbers(5, 3)
8 print(f"Sum: {sum_result}")
```

Variables Scopes

Global and Function Scope i

In Python, variables can have different scopes, including global and function scope. The scope of a variable defines where it can be accessed and modified.

Global Scope:

Variables declared outside of any function have a global scope. They can be accessed from anywhere in the code, both inside and outside functions.

```
1 global_var = 42
2
3 def a_func():
4     print(f"Inside the function, value = {global_var}")
5     print(f"ID of global_var: {id(global_var)}")
6
7 a_func()
8
9 print(f"Outside the function, value = {global_var}")
10 print(f"ID of global_var: {id(global_var)}")
```

Global and Function Scope ii

Function Scope:

Variables declared inside a function have a function scope. They are only accessible within the function where they are defined.

```
1 def a_func():
2     function_var = 99
3     print(f"Inside the function - function_var: {function_var}")
4
5 a_func()
6
7 # Uncommenting the next line would result in an error,
8 # as function_var is not accessible here.
9 # print(f"Outside the function - function_var: {function_var}")
```

Parameters scope of a Function:

In the example below, the global variable (`global_var`) and the function variable (`function_var`) are both same. You can check this by printing their ids.

```
1 def a_func(function_var):
2     print(f"Inside the function - id of function_var: {id(function_var)}")
3
4 global_var = 22
5 a_func(global_var)
6
7 print(f"Outside the function - id of global_var: {id(global_var)}")
```

Global and Function Scope iv

Scope of Parameters after Changing their Values:

In the example below, the global variable (`global_var`) and the function variable (`function_var`) will no longer remain the same when we change the value of the function variable. You can check this by printing their ids.

```
1 def a_func(function_var):
2     function_var = function_var + 1
3     print(f"Inside the function - id of function_var: {id(function_var)}")
4
5 global_var = 22
6 a_func(global_var)
7
8 print(f"Outside the function - id of global_var: {id(global_var)}")
```

But what about lists or any other mutable data structure?

Python Modules

Importing Modules i

In Python, a module is a file containing Python code. It can define functions, classes, and variables that can be reused in other Python programs. Modules help organize code into separate files, making it more manageable and maintainable.

Key Points about Modules:

- Modules are used to group related code together.
- They allow code reusability by importing functions and variables defined in other modules.
- In Python you can create your own custom modules.

Importing Modules ii

Example: Using a Module

Let's say we have a module named 'math_operations.py' that defines mathematical operations:

```
1 # math_operations.py
2 def add(a, b):
3     return a + b
4
5 def subtract(a, b):
6     return a - b
```

We can use this module in another Python files as follows:

```
1 import math_operations
2
3 result_add = math_operations.add(5, 3)
4 result_subtract = math_operations.subtract(10, 4)
5
6 print("Addition Result: {result_add}")
7 print("Subtraction Result: {result_subtract}")
```

Importing Modules iii

Also, you can import specific functions or variables from a module using the `from` keyword. There are two common ways to import from a module: importing specific functions and using a wildcard (`*`) import.

1. Importing Specific Functions:

```
1 from math_operations import add
2
3 result = add(5, 3)
4 print("Addition Result: {result}")
```

2. Using a Wildcard Import:

```
1 from math_operations import *
2
3 result_add = add(5, 3)
4 result_subtract = subtract(10, 4)
5
6 print("Addition Result: {result_add}")
7 print("Subtraction Result: {result_subtract}")
```


Problem Solving Approach with Examples

Problem Solving: Palindrome Checker

Problem Statement:

Write a program that prompts the user to enter a word and then determines whether the word is a palindrome.

- A palindrome is a word, phrase, number, or other sequence of characters that reads the same forward and backward, ignoring spaces, punctuation, and capitalization. In other words, a palindrome remains unchanged when its characters are reversed.

- Examples of palindromic words and phrases include:

radar, level, madam

Explanation: `''.join(reversed(input_string))`

- `input_string`: This is the original string that you want to reverse.
- `reversed(input_string)`: The `reversed` function takes an iterable (in this case, `input_string`) and returns a reversed iterator. It doesn't modify the original string but provides an iterator that yields the characters of the string in reverse order.
- `''.join(...)`: The `join` method is called on an empty string `''`. It takes an iterable (in this case, the reversed iterator) and concatenates its elements into a single string, using the empty string as the separator. Since there's no separator (empty string), it joins the characters without any spaces or characters in between.

So, `''.join(reversed(input_string))` effectively takes the characters of `input_string`, reverses their order using the `reversed` function, and joins them back together into a single string with no spaces or separators. This results in the reversed version of `input_string`.

Problem Solving: Fibonacci series i

Problem Statement: Write a Python program that generates the Fibonacci series up to a specified number of terms.

Fibonacci Series: The Fibonacci series is a sequence of numbers where each number (except the first two) is the sum of the two preceding ones. It starts with 0 and 1.

Example: The Fibonacci series with the first 10 terms looks like this:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Hint: You can use a loop to solve this problem.

Problem Solving: Primes i

Problem Statement: Write a Python program that generates a list of prime numbers up to a specified number.

Prime numbers: A Prime number is one that only divides on itself and 1.

Example: Below are the first 10 prime numbers:

1, 2, 3, 5, 7, 11, 13, 17, 19, 23 . . .