Python Lecture 4: Functions and Modules

September 11, 2025

1 Functions in Python

Functions are reusable blocks of code that perform a specific task. They help organize code, reduce repetition, and improve readability. A function can take input (parameters), process it, and optionally return output. Functions promote modularity, allowing complex programs to be broken into smaller, manageable parts.

1.1 Syntax for Defining and Calling Functions

To define a function, use the `def` keyword followed by the function name, parameters in parentheses, and a colon. The body is indented.

Structure:

```python
def function_name(parameters):
    # code to execute
    return value  # optional
```

To call a function, use its name followed by parentheses with arguments.

Example: A simple function to add two numbers.

```python
def add(a, b):
    return a + b

num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
result = add(num1, num2)
print(result)
```

Example Input: 5, 3
Output: 8.0

2 Types of Functions

2.1 Built-in Functions vs User-Defined Functions

Built-in functions are pre-defined in Python or libraries (e.g., print(), len()) and are ready to use without definition. User-defined functions are created by the programmer using `def` to perform custom tasks.

To use built-in functions from libraries, import the library first. For example, to use the math library, write 'import math'. You can then call functions like math.sqrt(16), which returns 4.0. Other libraries include: - pandas: Import with ’import pandas as pd’, and use pd.DataFrame([[1, 2], [3, 4]]) to create a DataFrame. - numpy: Import with ’import numpy as np’, and use np.array([1, 2, 3]) to create an array. - sklearn: Import with ’from sklearn.linear$_m odelimportLinearRegression', anduseLinearRegression()tocreateamodel. - keras : Importwith' fromkeras.modelsimportSequential', anduseSequential()tocreateaneuralnetworkmodel.$

Examples of calling built-in functions from the math library: - math.pow(2, 3) returns 8.0 (2 raised to 3). - math.sin(math.pi / 2) returns 1.0 (sine of 90 degrees). To call these, import math first, then use math.function$_n ame(arguments)$.

2.2 Types of User-Defined Functions

User-defined functions can be classified into four types based on parameters and return values. Here are examples using the task of adding three numbers (a, b, c) and optionally returning or printing the sum:

- Takes parameters, returns value: This function accepts inputs and returns a result. Define it as:

```python
def add(a, b, c):
    return a + b + c

a = float(input("Enter first number: "))
b = float(input("Enter second number: "))
c = float(input("Enter third number: "))
sum = add(a, b, c)
print(sum)
```

Call it with sum = add(1, 2, 3) and print(sum) to get 6.0.

- Takes parameters, no return: This function accepts inputs but only prints the result without returning it. Define it as:

```python
def print_sum(a, b, c):
    print(a + b + c)

a = float(input("Enter first number: "))
b = float(input("Enter second number: "))
c = float(input("Enter third number: "))
print_sum(a, b, c)
```

Call it with print$_s um(1, 2, 3) toget 6.0 printed.$

- No parameters, returns value: This function has no inputs and returns a computed value. Define it as:

```python
def get_sum():
    return 1 + 2 + 3

sum = get_sum()
print(sum)
```

Call it with sum = get$_s um() andprint(sum) toget 6.$

- No parameters, no return: This function has no inputs or return, performing an action. Define it as:

```python
def display_sum():
    print(1 + 2 + 3)

display_sum()
```

Call it with display$_sum()toget6printed.$

2.3 Another Example: Rectangle Area Computation

Using the four types for computing rectangle area (length * width).

- Takes parameters, returns value:

```python
def area(length, width):
    return length * width

length = float(input("Enter length: "))
width = float(input("Enter width: "))
rect_area = area(length, width)
print(rect_area)
```

Example Input: 5, 3
Output: 15.0

- Takes parameters, no return:

```python
def print_area(length, width):
    print(length * width)

length = float(input("Enter length: "))
width = float(input("Enter width: "))
print_area(length, width)
```

Example Input: 5, 3
Output: 15.0

- No parameters, returns value:

```python
def get_area():
    length = float(input("Enter length: "))
    width = float(input("Enter width: "))
    return length * width

rect_area = get_area()
print(rect_area)
```

Example Input: 5, 3
Output: 15.0

- No parameters, no return:

```python
def display_area():
    length = float(input("Enter length: "))
    width = float(input("Enter width: "))
    print(length * width)

display_area()
```

Example Input: 5, 3
Output: 15.0

2.4 Additional Function Examples

2.4.1 Function to Decide if a Number is Even or Odd

Write a function that takes a number and returns whether it is even or odd.

.

```python
def is_even_odd():
    num = float(input("Enter a number: "))
    if num % 2 == 0:
        return "Even"
    else:
        return "Odd"

result = is_even_odd()
print(result)
```

Example Input: 4
Output: Even

2.4.2 Function for Computing the Grade

Write a function that takes a score and returns the grade based on the score range.

.

```python
def compute_grade():
    score = float(input("Enter score (0-100): "))
    if score >= 90:
        return "A"
    elif score >= 80:
        return "B"
    elif score >= 70:
        return "C"
    else:
        return "Fail"

grade = compute_grade()
print(grade)
```

Example Input: 85
Output: B

2.4.3 Function to Add Even Numbers

Write a function that takes a list of numbers and returns the sum of even numbers.

.

```python
def sum_even_numbers():
    numbers = input("Enter numbers separated by spaces: ").split()
    total = 0
    for num in numbers:
        num = float(num)
        if num % 2 == 0:
            total += num
    return total

even_sum = sum_even_numbers()
print(even_sum)
```

Example Input: 1 2 3 4 5 6
Output: 12.0