



Python Functions

Functions Basics

in Python



SQL

PYTHON

JAVA

HTML

C#

What's in it for you?

- ▶ **What is a Function in Python?**
- ▶ **Types of Functions**
- ▶ **Advantages of using Functions**

What is a Function in Python?

- A function is a bunch of related statements that are defined to perform a specific task.
- Functions helps fragment our program into smaller and modular chunks.
- Functions allows you to make extensive programs more organized and manageable.
- It also avoids repetition and makes the code reusable.



Types of Python Functions

Built-in functions

Functions that are built into Python



User-defined functions

Functions defined by users themselves

Built-in Python Functions

Built-in functions are pre-defined functions in Python

pow()

abs()

print()

round()

len()



range()

next()

max()

slice()

sorted()

User-defined functions

These functions are defined by the user to perform a specific task

Steps to create a user-defined function

- Creating a function
- Calling a function



Syntax:

```
def my_function(parameters):  
    function_block  
return expression
```

Example:

```
def my_function():  
    print("Hello Python")
```

Output:

Hello Python

- With the help of functions, we can avoid rewriting the same logic or code again and again in a program
- In a single program, we can call Python functions anywhere multiple times
- We can track an extensive Python program easily when it is divided into numerous functions
- The main achievement of Python functions is its Reusability



```
[1]: def func1():  
      print("Hello MAC108")  
      func1()
```

Hello MAC108

```
[2]: def ft_to_cm(ft):  
      cm = ft * 30.48  
      return cm  
      ft_to_cm(10)
```

```
[2]: 304.8
```

```
[3]: def add(a, b):  
      return a + b  
      add(5, 10)
```

```
[3]: 15
```

```
[4]: def mult(x, y):  
      result = x * y  
      print(result)  
      mult(5, 10)
```

```
50
```

```
[5] : def absolute(x):  
      if x >= 0:  
          return x  
      else:  
          return -x  
absolute(10)
```

```
[5] : 10
```

```
[6] : def absolute(x):  
      if x >= 0:  
          return x  
      else:  
          return -x  
absolute(-20)
```

```
[6] : 20
```



```
[8]: def triangle_area(base, height):  
      """  
      Calculate the area of a triangle.  
  
      Parameters:  
      - base (float): The length of the base of the triangle.  
      - height (float): The height of the triangle.  
  
      Returns:  
      float: The area of the triangle.  
      """  
      area = 0.5 * base * height  
      return area  
  
# Example usage:  
base_length = 5.0  
triangle_height = 8.0  
  
result_area = triangle_area(base_length, triangle_height)  
print(f"The area of the triangle is: {result_area}")
```

The area of the triangle is: 20.0

```
[9]: def solve_linear_equation(a, b):  
      """  
      Solve a linear equation of the form  $ax + b = 0$ .  
  
      Parameters:  
      - a (float): Coefficient of the variable x.  
      - b (float): Constant term.  
  
      Returns:  
      float: The solution for x.  
      """  
      if a == 0:  
          if b == 0:  
              return "Infinite solutions (identity equation)"  
          else:  
              return "No solution (contradictory equation)"  
      else:  
          x = -b / a  
          return x  
  
      # Example usage:  
      coeff_a = 2.0  
      constant_b = -4.0  
  
      solution = solve_linear_equation(coeff_a, constant_b)  
      print(f"The solution for x is: {solution}")  
  
The solution for x is: 2.0
```

```
[10]: def sphere_diameter(radius):  
      """  
      Calculate the diameter of a sphere given its radius.  
  
      Parameters:  
      - radius (float): The radius of the sphere.  
  
      Returns:  
      float: The diameter of the sphere.  
      """  
      diameter = 2 * radius  
      return diameter  
  
# Example usage:  
sphere_radius = 3.0  
  
result_diameter = sphere_diameter(sphere_radius)  
print(f"The diameter of the sphere is: {result_diameter}")
```

The diameter of the sphere is: 6.0

```
[11]: # Example usage: Earth diameter  
sphere_radius = 7917.5  
  
result_diameter = sphere_diameter(sphere_radius)  
print(f"The diameter of the sphere is: {result_diameter}")
```

The diameter of the sphere is: 15835.0


```
[12]: import math

def earth_circumference():
    """
    Calculate the Earth's circumference.

    Returns:
    float: The circumference of the Earth.
    """
    # Earth's mean radius in kilometers
    earth_radius = 6371.0

    # Calculate circumference using the formula  $C = 2 * \pi * r$ 
    circumference = 2 * math.pi * earth_radius
    return circumference

# Example usage:
result_circumference = earth_circumference()
print(f"The Earth's circumference is approximately: {result_circumference:.2f} kilometers")
```

The Earth's circumference is approximately: 40030.17 kilometers







