Python Functions

Functions are essential in Python for code organization and reusability. Learn how to create and use functions effectively in this presentation.

by Harish Kumar Sharma

```
_mod = modifier_ob.modifiers.new("
 object to mirror_ob
_mod.mirror_object = mirror_ob
tion == "MIRROR_X":
_mod.use_x = True
_mod.use_y = False
_mod.use_z = False
ation == "MIRROR_Y":
_mod.use_x = False
_mod.use_y = True
_mod.use_z = False
ation == "MIRROR_Z":
_mod.use_x = False
_mod.use_y = False
_mod.use_z = True

tion at the end -add back the desel
.select= 1
ob.select=1
t.scene.objects.active = modifier_ob
lected" + str(modifier_ob)) # modif
_ob.select = 0
.context.selected_objects[0]
.objects[one.name].select = 1

("please select exactly two objects,

OPERATOR CLASSES ----------------

.Operator):
 mirror to the selected object"""
.mirror_mirror_x"
x"

ject is not None
```

Why Are Functions Important?

1 Code Organization

Functions help break down complex code into smaller, manageable parts.

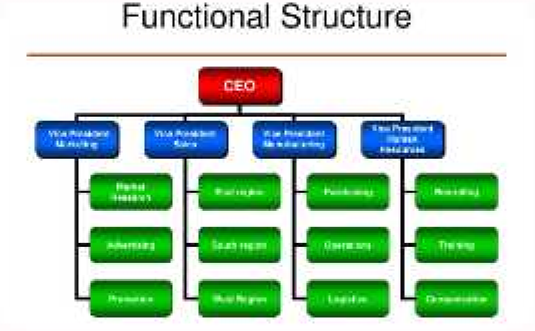
2 Code Reusability

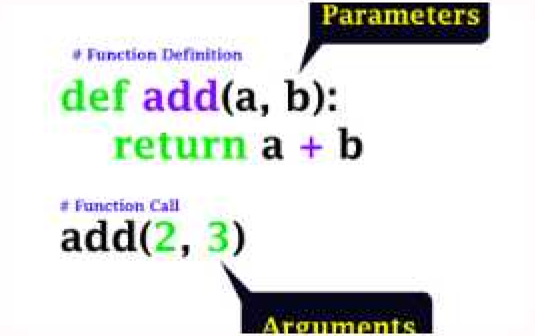
Functions can be reused multiple times, saving time and effort.

3 Modularity

Functions promote modular programming, making code easier to understand and maintain.

Syntax and Structure





```
# Function Definition
def add(a, b):
    return a + b

# Function Call
add(2, 3)
```

Syntax

Learn the proper syntax for defining and calling functions in Python.

Structure

Understand the structure of a function and the purpose of each component.

Parameters & Arguments

Explore how to pass values to functions and retrieve them using parameters and arguments.

The Return Statement

1 Returning Values

Discover how to use the return statement to return values from a function.

2 Multiple Returns

Learn how to return multiple values from a function using tuples or other data structures.

3 Void Functions

Understand void functions that do not return any values, but perform other operations.

Scope and Variables

Global Scope

Explore global variables and their accessibility throughout your program.

Local Scope

Dive into local variables and understand their limited visibility within functions.

Variable Shadowing

Discover the concept of variable shadowing and how it affects variable lookup.

Built-in Functions

1

Math Functions

Learn about mathematical functions like abs(), pow(), max(), min(), and more.

2

String Functions

Explore powerful string manipulation functions, such as len(), upper(), lower(), and split().

3

List Functions

Discover functions like append(), sort(), reverse(), and count() for manipulating lists.

4

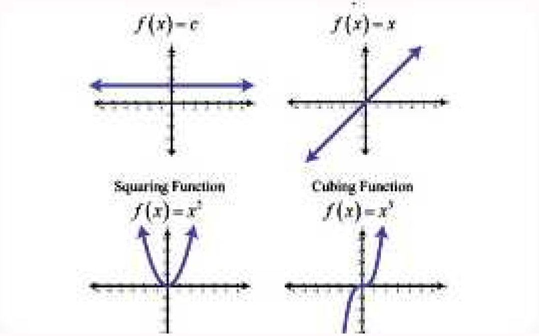
Other Categories

Get introduced to functions for file manipulation, input/output, and datetime operations.

Creating and Using User-Defined Functions

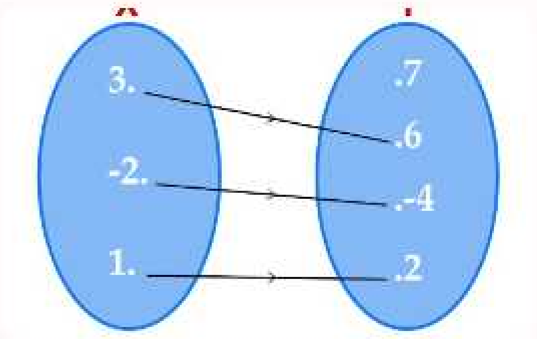
Create Functions

Learn how to define your own custom functions tailored to your specific needs.



Function Calling

Discover different ways of calling user-defined functions and passing arguments.



Real-World Examples

Explore examples of user-defined functions and the real-world problems they can solve.

Conclusion

Python functions are an essential tool in any programmer's arsenal. By understanding their purpose, syntax, and best practices, you can elevate your coding skills to the next level.