# Functions and Data Structures in Python

Python is a powerful language for data manipulation and analysis. This presentation introduces core concepts, including functions and data structures, that are essential for writing effective Python code.

by Adamya Neupane

# Writing Functions: Syntax and Best Practices

Functions are reusable blocks of code that perform specific tasks. They follow a defined syntax, including the function name, parameters, and the code block. Best practices involve writing clear, concise functions that are easy to understand and maintain.

## Function Definition

To define a function, you use the `def` keyword, followed by the function name, parentheses containing any parameters, and a colon.

## Function Body

The function body contains the code that the function executes. It is indented to distinguish it from the rest of the code.

## Docstrings

Docstrings are used to document the purpose and usage of a function. They are enclosed within triple quotes.

# Function Parameters and Return Values

Functions can take input values, called parameters, that are used within the function's code. A function can also return a value, which can be used by the code that called the function.

1

### Parameters

Parameters are variables that are passed into a function when it is called. They provide input to the function.

2

### Return Values

A function can return a value using the `return` keyword. This value can be used by the code that called the function.

3

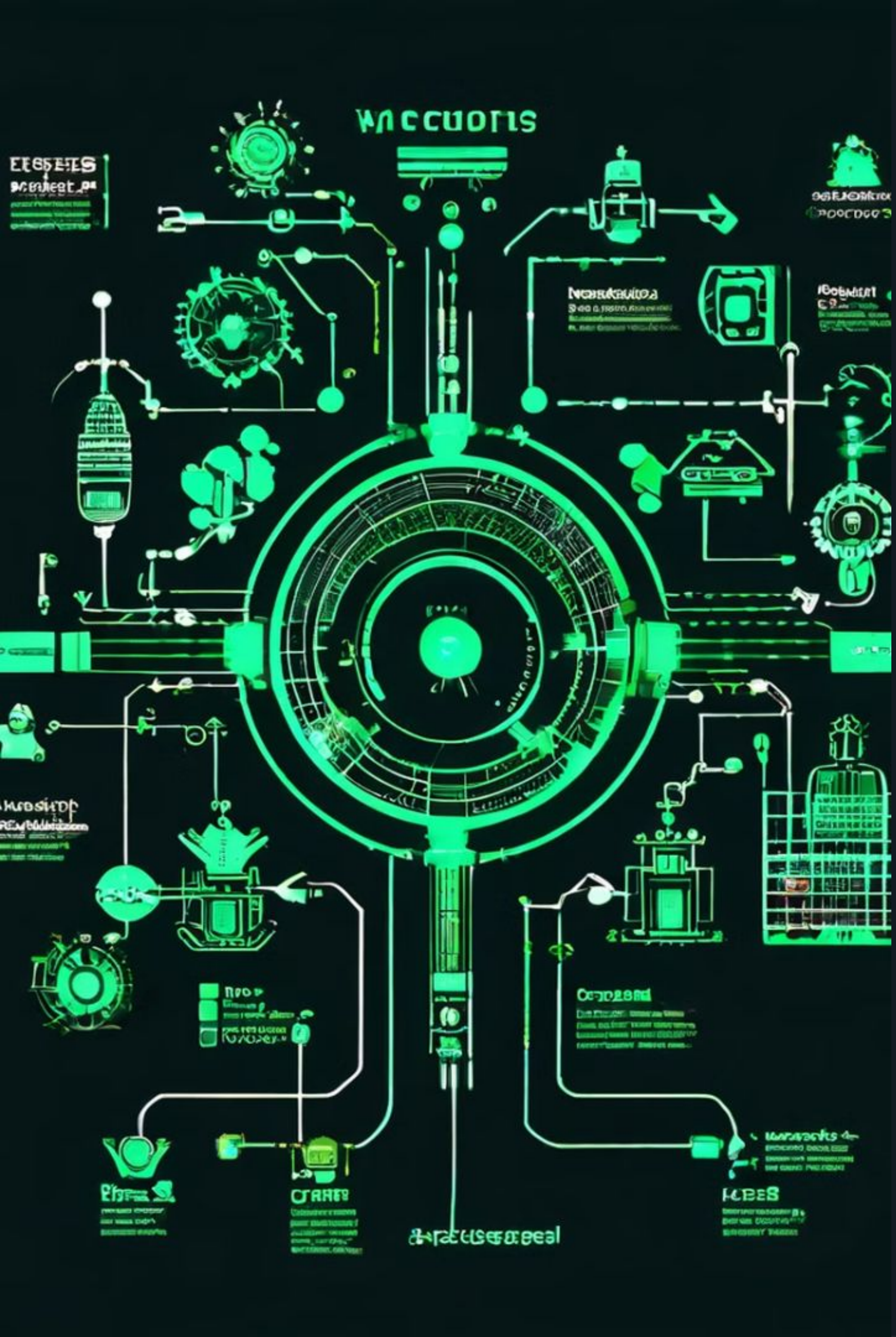
### Default Parameters

Functions can have default parameters that are assigned a value if no value is provided when the function is called.

4

### Keyword Arguments

Keyword arguments allow you to pass parameters to a function by name, making the order of arguments less important.



# Scope and Variable Lifetime in Functions

Variables in Python have different scopes depending on where they are defined. A variable's lifetime refers to how long it exists in memory. Understanding scope and lifetime helps you write functions that behave as expected.

1

## Global Scope

Variables defined outside any function have global scope and are accessible from anywhere in the program.

2

## Local Scope

Variables defined inside a function have local scope and are only accessible within that function.

3

## Lifetime

Local variables are created when a function is called and destroyed when the function completes.

# Built-in Data Structures: Lists, Tuples, and Dictionaries

Python provides several built-in data structures that are used to organize and store collections of data. These structures offer different features and use cases, allowing you to choose the best option for your needs.

## Lists

Ordered, mutable collections of items. Elements can be of different data types.

## Tuples

Ordered, immutable collections of items. Elements can be of different data types.

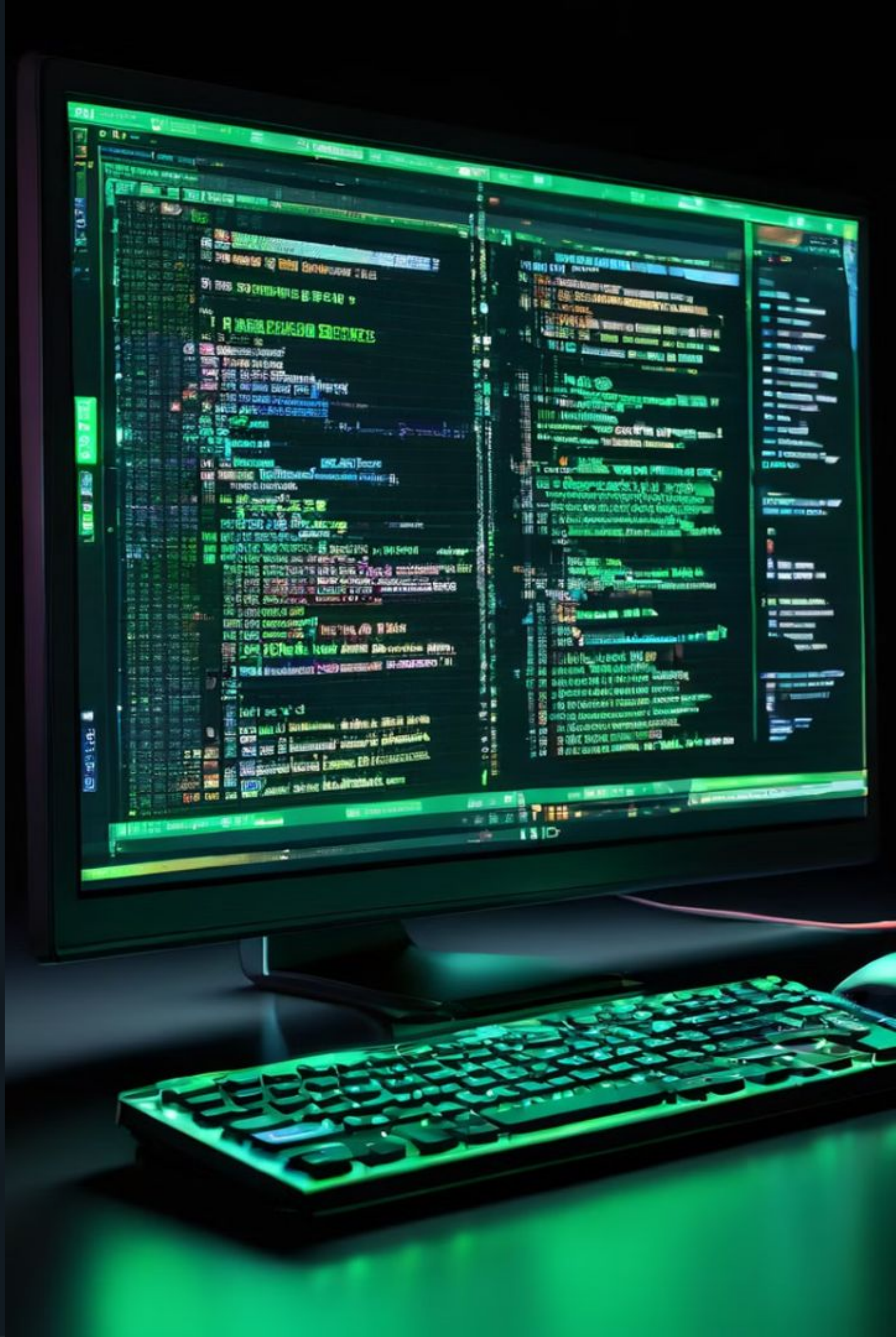
## Dictionaries

Unordered collections of key-value pairs. Keys must be unique and immutable, while values can be of any data type.

# Accessing and Manipulating Data Structures

Data structures allow you to store and organize data effectively. You can access and modify elements within data structures using various methods and operators, depending on the specific structure.

| Data Structure | Access | Modification |
|---|---|---|
| List | Index-based access (e.g., `list[0]`) | Append, insert, remove, replace |
| Tuple | Index-based access (e.g., `tuple[0]`) | Immutable, cannot be modified |
| Dictionary | Key-based access (e.g., `dict['key']`) | Add, update, delete key-value pairs |



# Iterating over Data Structures

Iteration allows you to process each element in a data structure individually. Python provides loop structures like `for` and `while` to iterate over data structures.

### For Loop

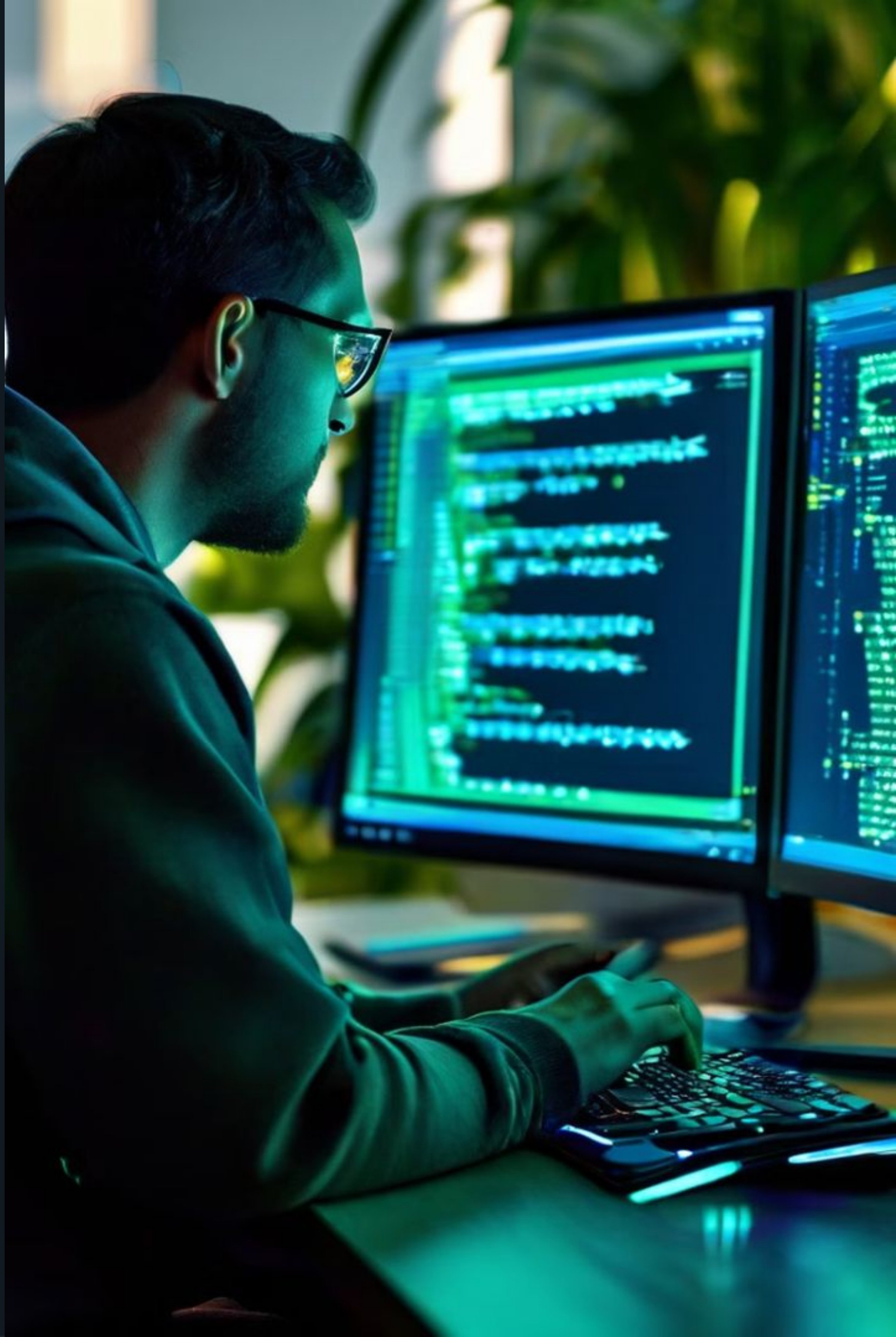
Iterates over each element in a sequence, executing a block of code for each element.

### List Comprehension

Provides a concise way to create new lists based on existing lists or other iterable objects.

### Dictionary Iteration

Iterates over key-value pairs in a dictionary, allowing access to both the keys and values.



# Nested Data Structures and Comprehensions

Nested data structures allow you to organize data in a hierarchical manner. Comprehensions provide concise ways to create new data structures from existing ones.

1

## Lists of Lists

A list can contain other lists, creating a nested structure. This is useful for representing multi-dimensional data or relationships.

2

## Dictionaries with Lists

Dictionaries can store lists as values. This allows you to organize data based on keys and store related information within lists.

3

## Comprehensions

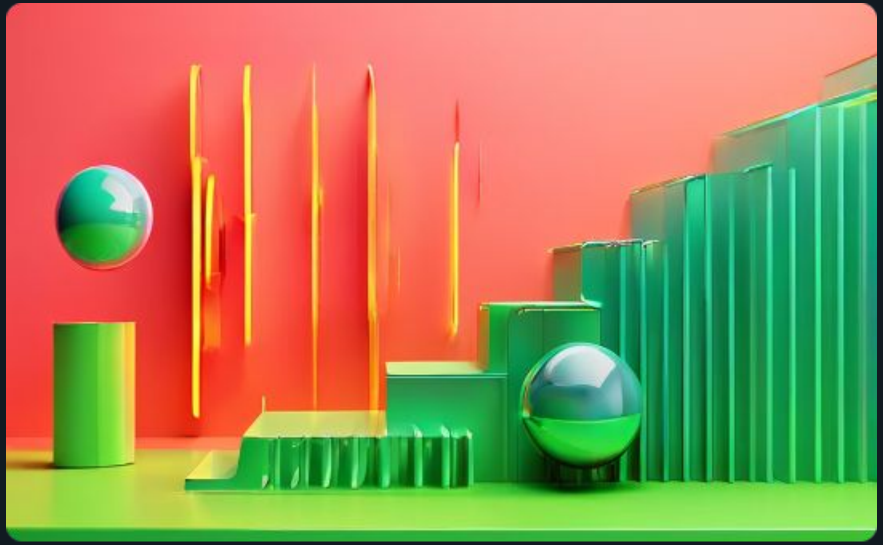
Comprehensions offer a concise way to create new lists, sets, or dictionaries from existing data. They can be used for filtering, mapping, and other transformations.

# Choosing the Right Data Structure for Your Needs

Selecting the most appropriate data structure for your needs is essential for efficient and effective code. Consider the characteristics of each data structure and how they align with the task you need to perform.

### Lists

Use for storing ordered collections of elements, such as a list of items to buy or a sequence of tasks.



### Sets

Use for storing unique and unordered collections of elements, such as a set of distinct words or unique user IDs.

### Dictionaries

Use for storing key-value pairs, such as storing student names as keys and their grades as values.

# Conclusion and Additional Resources

Functions and data structures are fundamental building blocks for Python programming. By mastering these concepts, you can write more efficient, readable, and maintainable code. Explore additional resources like online documentation and tutorials to deepen your understanding.

1

## Practice

Practice using functions and data structures in various coding exercises and projects to solidify your knowledge.

2

## Documentation

Refer to the official Python documentation for comprehensive information on functions, data structures, and other language features.

3

## Online Resources

Explore platforms like Codecademy, freeCodeCamp, and W3Schools for interactive tutorials and code examples.