



Python Basics

Python is a versatile and powerful programming language. It's used in various fields, from web development to data science.



by **Adamy
Neupane**



Print Statements

Print statements are used to display output on the console. They're essential for interacting with the user and displaying results.

1

Syntax

The basic syntax is ``print("your text here")``.

2

Example

The code ``print("Hello, World!")`` will display "Hello, World!" on the console.

Data Types

Data types define the kind of values a variable can hold. Understanding these is crucial for writing correct Python code.

Numeric

Represent numbers, such as integers and floating-point numbers.

- Integers
- Floats

Text

Represent sequences of characters, like words and sentences.

- Strings

Logical

Represent truth values, either ``True`` or ``False``.

- Booleans



Integers

Integers represent whole numbers without decimal points. They're used for counting and calculations.

Example

The number `10` is an integer.

Operations

You can perform basic arithmetic operations on integers, like addition, subtraction, multiplication, and division.



Floats

Floats represent numbers with decimal points. They're useful for representing measurements and more precise calculations.

1

Definition

A float is a number with a decimal point, like 3.14.

2

Operations

Floats support the same operations as integers, including division and modulo.

3

Example

The code ``print(3.14 * 2)`` will output ``6.28``.

Strings

Strings are sequences of characters, like letters, numbers, and symbols. They're used to store and manipulate text data.

Syntax

Strings are enclosed in single or double quotes.

Example

`"Hello, world!"` or `'This is a string.'`

Operations

You can concatenate strings, extract substrings, and convert them to uppercase or lowercase.



Booleans

Booleans represent truth values, either `True` or `False`. They're used for logical comparisons and conditional statements.



True

Indicates a condition is met.



False

Indicates a condition is not met.



Lists

Lists are ordered collections of items. They're mutable, meaning you can change their contents after creation.

1

Creation

Lists are created using square brackets `[]` and elements separated by commas.

2

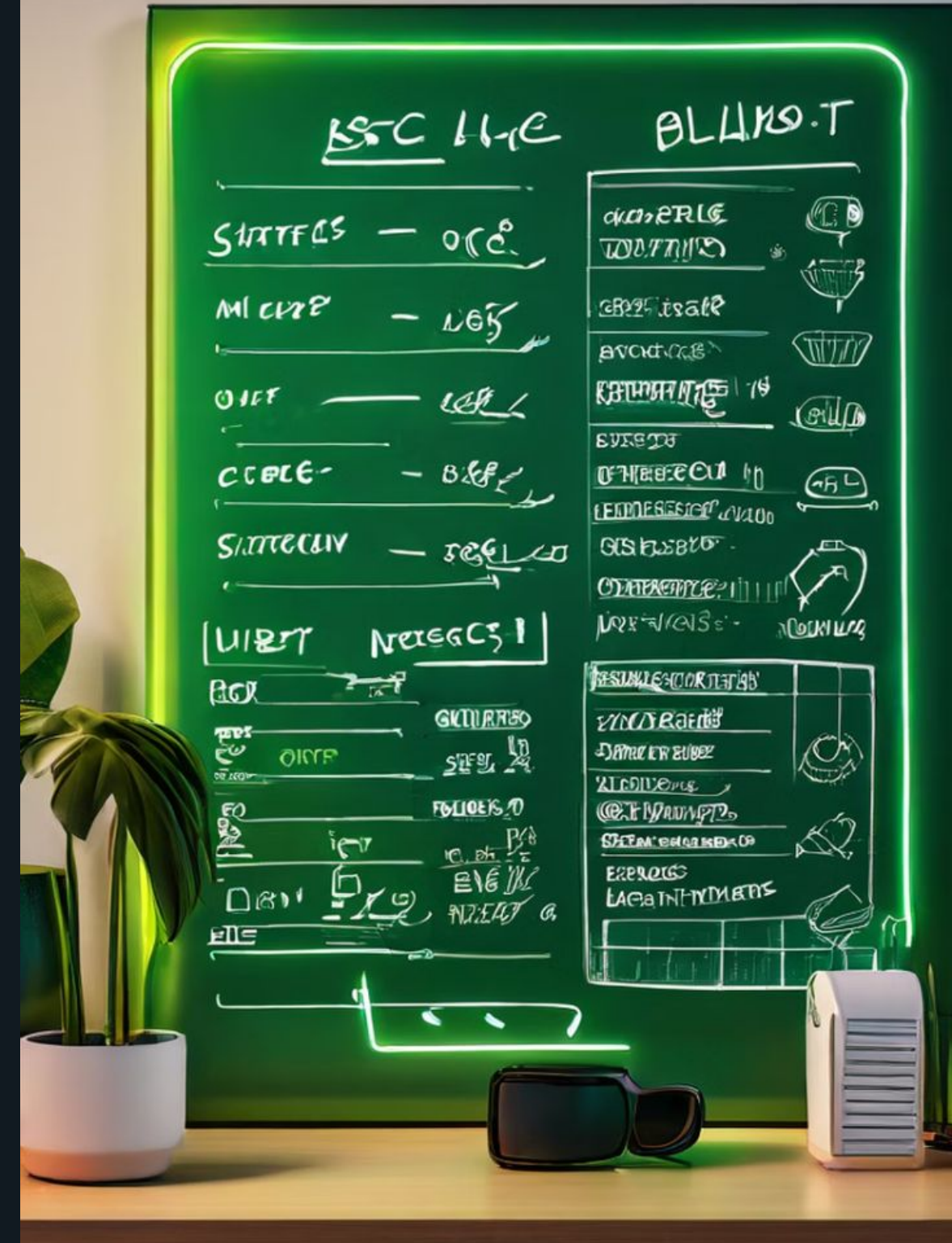
Access

Elements are accessed using their index, starting from 0.

3

Modification

You can add, remove, or modify elements using methods like `append`, `remove`, and `insert`.



Tuples

Tuples are ordered collections of items, similar to lists. However, they are immutable, meaning you cannot change their contents after creation.

Empty Tuple

An empty tuple is created using `()`.

Single-Element Tuple

A single-element tuple requires a comma after the element, like `(1,)`.



Dictionaries

Dictionaries are unordered collections of key-value pairs. They allow you to store and access data using meaningful keys.

Creation

Dictionaries are created using curly braces `{}` and key-value pairs separated by colons `:`.

Access

Values are accessed using their corresponding keys.

Modification

You can add, remove, or modify key-value pairs using methods like `update`, `pop`, and `del`.

Taking User Inputs



Flexible Input

Python's `input()` function allows users to enter any type of data, from text to numbers, giving your programs dynamic functionality.



Interactive Experience

By prompting users for input, you can create an engaging, personalized experience that responds to their needs and preferences.



Data Validation

You can use conditional statements to validate user input and ensure your program handles data correctly, preventing errors.

Taking User Inputs and Performing Operations

In this example, we'll demonstrate how to take user input and perform basic operations on it. The `input()` function in Python allows users to enter any type of data, from text to numbers, giving your programs dynamic functionality and an interactive user experience.

First, we'll prompt the user to enter a number. We'll then square that number and display the result. This shows how you can use user input to drive program logic and perform calculations.