NAME: Nishant

DATE: 16 SEPTEMBER - 2024

INTRODUCTION TO PYTHON

HISTORY OVERVIEW

CHRONOLOGICAL DEVELOPMENT

Python was created by Guido van Rossum and first released in 1991. It was designed to emphasize code readability and simplicity. Over the years, language features and syntax have evolved, leading to new versions. The introduction of Python 2.0 and later Python 3.0 marked key shifts in the language's functionality and usability.

CORE FUNCTIONS

CONTROL STRUCTURES

Control structures such as loops and conditionals govern the flow of execution in Python. They include 'if', 'for', and 'while' statements which enhance code flexibility. Mastering control structures enables more complex program functionalities. Implementing these structures is essential for problem-solving in programming.

BASIC DATA TYPES

Python supports various data types including integers, floats, strings, and booleans. This versatility allows developers to handle different kinds of data efficiently. Understanding these core types is critical for programming tasks. Each data type has unique methods and operations that can be utilized in code.

MODULES IN PYTHON

IMPORTANCE OF MODULES

[illegible]

USING MODULES EFFECTIVELY

Python has a rich ecosystem of modules available through its Standard Library. Importing modules is straightforward, allowing for quick integration of various functionalities. Additionally, third-party modules can be installed using package managers like pip, extending Python's capabilities. This flexibility makes Python an essential tool for development.

HISTORICAL MILESTONES

YEAR	MILESTONE	DESCRIPTION
1991	First Release	Official release of Python 1.0.
2000	Python 2.0	Introduction of list comprehensions and garbage collection.
2008	Python 3.0	Major redesign of the language that emphasized removing redundant features.
2020	End of Life for Python 2	Discontinuation of support for Python 2.x versions.

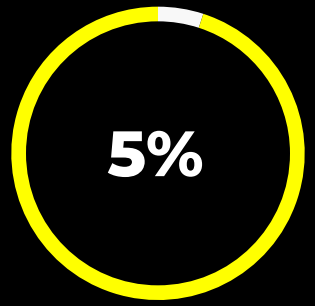
FUNCTION USAGE STATISTICS

STRING FUNCTIONS

String functions are used in 50% of Python programs due to the importance of text manipulation across applications. They handle tasks such as formatting and searching strings efficiently.

FILE HANDLING FUNCTIONS

File handling functions comprise 15% of usage statistics. They enable reading from and writing to files, which is crucial for data persistence in programs.



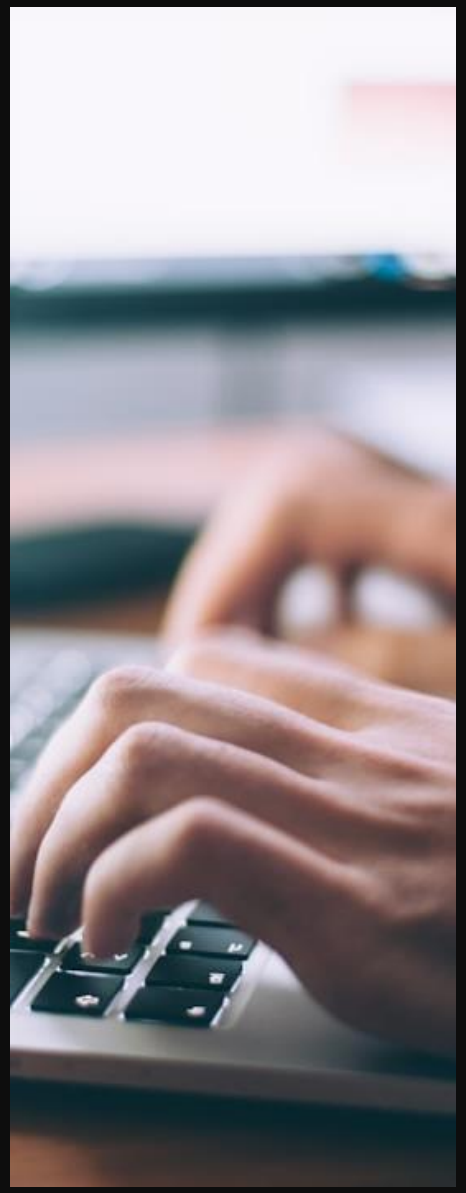
DICTIONARY FUNCTIONS

Dictionary functions are utilized in about 5% of cases, showing the specific needs for key-value pair storage in Python applications. They are essential for certain data structures.

LIST FUNCTIONS

List functions account for 30% of usage, highlighting their role in managing collections of data. Lists are vital in data processing and manipulation tasks.

FUTURE OF PYTHON



FUTURE DEVELOPMENTS

The future of Python looks promising with ongoing enhancements focusing on performance and ease of use. New libraries and frameworks are emerging, catering to areas like artificial intelligence and data science. Community support continues to grow, fostering innovation and collaboration among developers. As Python evolves, its applicability across various domains is expanding significantly.

FUNCTION AND MODULES BASICS

KEY FUNCTION CONCEPTS

FUNCTION SCOPE & LIFESPAN

Each function has a specific scope and lifespan, determining where its variables can be accessed. Local variables exist only within the function, while global variables can be accessed anywhere in the code. Understanding this distinction prevents variable conflicts and promotes better resource management.

DEFINITION OF FUNCTIONS

Functions are reusable blocks of code that perform a specific task. They help in organizing code, making programs easier to understand and manage. Functions can take inputs, known as arguments, and return outputs. Mastering function definitions is essential to writing efficient code.

MODULES OVERVIEW

MODULE TYPE	DESCRIPTION	EXAMPLES
Standard Libraries	Built-in modules providing core functionality.	math, datetime, sys
External Libraries	Additional modules created by the community.	NumPy, Pandas, Requests
User-Defined Modules	Custom modules created by developers.	Custom functions saved in .py files