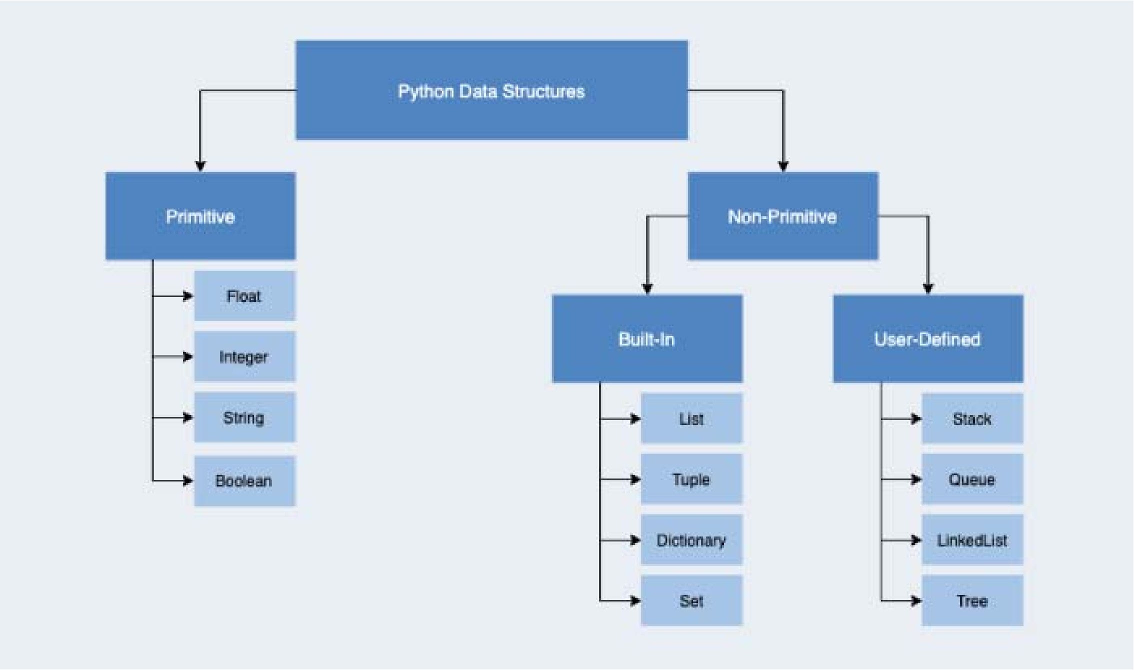
Data Structure Overview

Data structures are fundamental concepts of computer science which helps is writing efficient programs in any language. Python is a high-level, interpreted, interactive and object-oriented scripting language using which we can study the fundamentals of data structure in a simpler way as compared to other programming languages.

What is a Data Structure?

Organizing, managing and storing data is important as it enables easier access and efficient modifications. Data Structures allows you to organize your data in such a way that enables you to store collections of data, relate them and perform operations on them accordingly.



Follow : SIDDHESH Kumbhar LinkedIn: www.linkedin.com/in/siddhesh-kumbhar 517627b4

Types of Data Structures in Python

Python has implicit support for Data Structures which enable you to store and access data. These structures are called List, Dictionary, Tuple and Set.

Python allows its users to create their own Data Structures enabling them to have full control over their functionality. The most prominent Data Structures are Stack, Queue, Tree, Linked List and so on which are also available to you in other programming languages. So now that you know what are the types available to you, why don't we move ahead to the Data Structures and implement them using Python.

Built-in Data Structures

1. List: An ordered, mutable, and heterogeneous collection of elements. Lists are defined using square brackets [ ].
2. Tuple: An ordered, immutable, and heterogeneous collection of elements. Tuples are defined using parentheses ( ).
3. Dictionary: An unordered collection of key-value pairs, where keys must be unique and hashable. Dictionaries are defined using curly braces { }.
4. Set: An unordered collection of unique and hashable elements. Sets are defined using curly braces { } with elements separated by commas, or the built-in set( ) function.

These data structures are fundamental to Python programming and provide the means for organizing and manipulating data in your programs.

User Defined Data Structures in python

In Python, you can also create your own custom data structures, also known as user-defined data structures. Some common examples of user-defined data structures include:

1. Linked Lists: A data structure that consists of a series of nodes, where each node contains a reference to an object and a reference to the next node in the list.
2. Stacks: A data structure that follows the Last-In-First-Out (LIFO) principle, where the last element added to the stack is the first one to be removed.
   3.Queues: A data structure that follows the First-In-First-Out (FIFO) principle, where the first element added to the queue is the first one to be removed.
3. Trees: A hierarchical data structure where each node has a parent and zero or more child nodes.

By using user-defined data structures, you can create complex data structures that meet the specific requirements of your application, and abstract away the underlying implementation details, making your code more readable and maintainable.

Primitive Data Structures in Python

1. Integer: A whole number, for example, -5, 0, or 42.
2. Float: A real number with a decimal component, for example, 3.14 or -2.71828.
3. Boolean: A value that represents truth, either True or False.
4. String: An ordered sequence of characters, for example, "hello" or "python".
5. None: A special constant that represents the absence of a value.

These primitive data structures form the building blocks of more complex data structures in Python, such as lists, dictionaries, and sets, which are built using combinations of these basic data types.