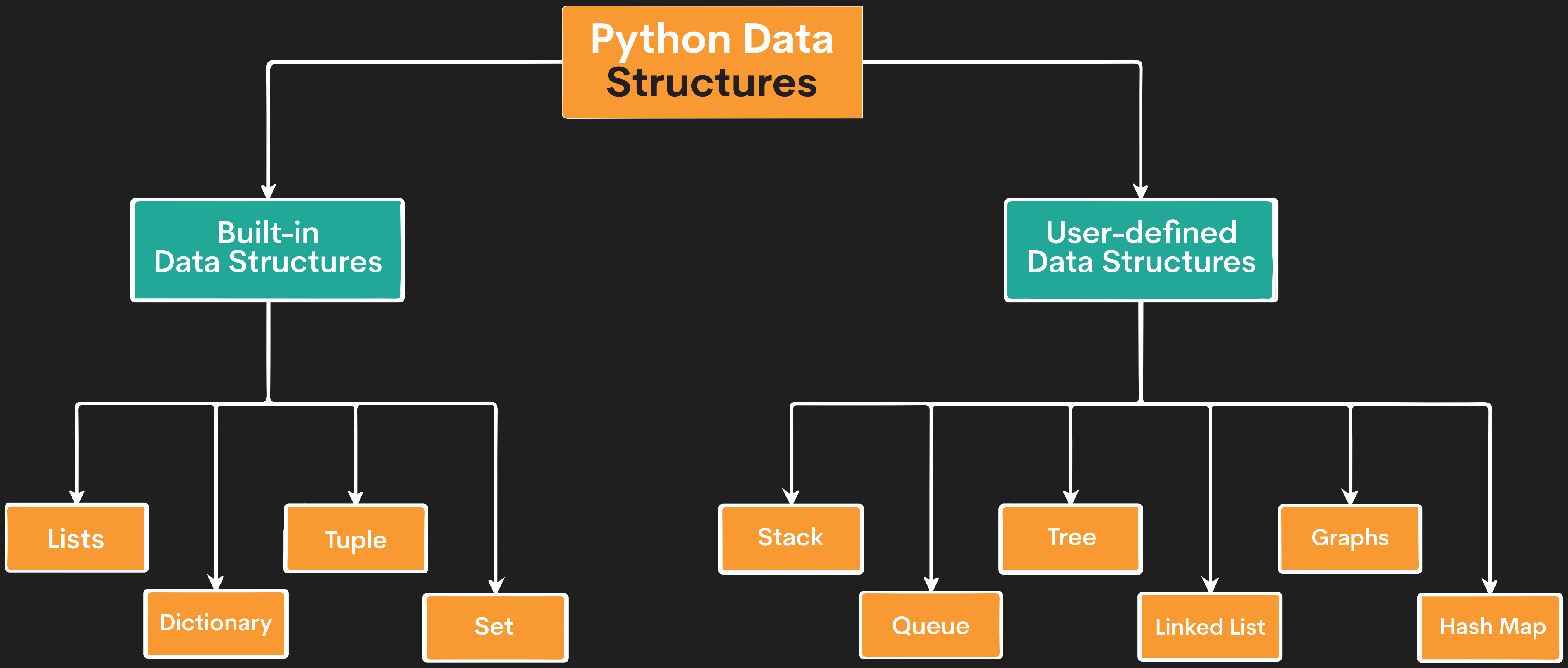
THE CHEAT SHEET FOR PYTHON DATA STRUCTURES

THE

CHEAT SHEET FOR PYTHON DATA STRUCTURES



BUILT IN DATA STRUCTURES

Lists

for storing multiple items in a single variable

- **Changeable**
  the data can be removed, added, or changed
- **Ordered**
  the data order is defined and unchanged
- **Duplicates**
  it can contain data of the same values

Dictionary

for storing values in the key-value pairs

- **Changeable**
  the data can be removed, added, or changed
- **Ordered**
  the data order is defined and unchanged
- **No Duplicates**
  it can't contain data of the same values

BUILT IN DATA STRUCTURES

Set

for storing multiple items
in a single variable

- **Unchangeable**
  the data can't be removed, added, or changed after creating the set
- **Unordered**
  the data order is not defined and will change with every use of the list
- **No Duplicates**
  it can’t contain data of the same values

Tuple

for storing multiple items
in a single variable

- **Unchangeable**
  the data can't be removed, added, or changed after creating the set
- **Ordered**
  the data order is defined and unchanged
- **Duplicates**
  it can contain data of the same values

USER-DEFINED DATA STRUCTURES

Stack

for storing and retrieving data sequentially,
e.g., as temporary storage of data within procedures

- **Linear data structure**
  data is arranged in a linear manner where every new element is linked to the previous and/or next element
- **Last In-First Out (LIFO) or First In-Last Out (FILO) method**
  adding a new element to one end and deleting it from the same end

Queue

for storing and retrieving data sequentially,
e.g., as a control of access to shared resources

- **Linear data structure**
  data is arranged in a linear manner where every new element is linked to the previous and/or next element
- **First In-First Out (FIFO) method**
  adding a new element to one end and deleting the element from the other end (the least recent element)

USER-DEFINED DATA STRUCTURES

(Binary) Tree

for storing and retrieving hierarchical data,
e.g., the organizational structure of a company

- **Hierarchical data structure**
  data is arranged hierarchically with data represented with nodes and children nodes, with each node holding a reference to every child node
- **Two children**
  each node has a maximum of two children (left and right)
- **Node reference ≥ right child node**
  a reference stored in the node is always equal to or greater than the reference stored in the left child node
- **Node reference ≤ left child node**
  a reference stored in the node is always equal to or less than the reference stored in the right child node

USER-DEFINED DATA STRUCTURES

Linked List

for storing and retrieving data sequentially in the form of nodes that contain its data and the address of the following node, e.g., dynamic memory allocation

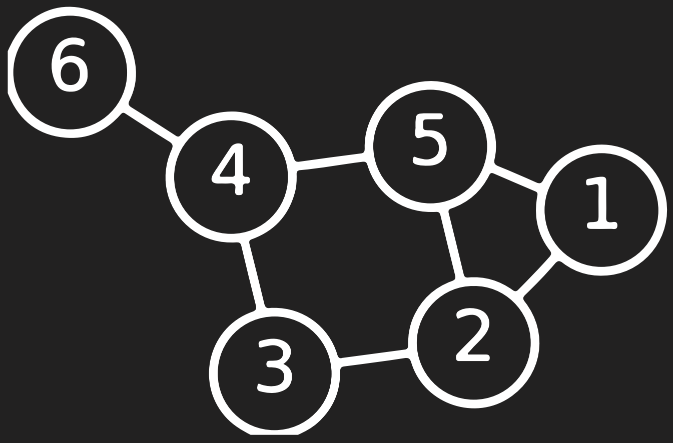
- **Linear data structure**
  data is arranged in a linear manner where data is linked by pointers
- **Randomness**
  nodes are stored randomly in the memory

Hash Map

for storing the data through the key-value pair and making data insertion, deletion, update, and retrieval quicker

- **Indexed data structure**
  maps the element's key or index value and calculates it using the hash function
- **Key-value pair**
  assigns each element a key-value pair

USER-DEFINED DATA STRUCTURES



Graph

for storing and retrieving data sequentially in the form of nodes that contain its data and the address of the following node, e.g., dynamic memory allocation

- **Linear data structure**
  data is arranged in a linear manner where data is linked by pointers
- **Randomness**
  nodes are stored randomly in the memory

SPECIALIZED DATA STRUCTURES

namedtuple()	Gives a descriptive name to each position in the tuple and is used for accessing values instead of indices.
deque	A double-ended queue where elements can be added or removed from both left and right sides.
ChainMap	Groups multiple dictionaries and other mappings to create a single updateable view.
Counter	A dictionary subclass that counts hashable objects storing them as keys andcounting them as values.
OrderedDict	A dictionary subclass that keeps the order in which the items are inserted into the dictionary.
defaultdict	A dictionary subclass for assigning each new key with a default value based on the dictionary type.
UserDict	A class that simulates the dictionary and simplifies dictionary subclassing.
UserList	A class that simulates the list and simplifies list subclassing.
UserString	A class that simulates the string and simplifies string subclassing.

follow us!

For more data science tips

@stratascratch

STRATASCRATCH

www.stratascratch.com