Introduction to Python Data Structure

Python Data Structure

Difference between List, Tuple & set

Example: Dictionary

Example: List

Python Data Structure

Python provides several built-in data structures that allow you to store and organize data efficiently. Some of the most commonly used data structures in Python are:

Lists:

- A collection of items separated by commas and enclosed in square brackets [].
- Lists are mutable, meaning you can change their contents.
- Example:

  ```
  my_list = ["apple", "banana", "cherry"]
  my_list.append("orange")
  ```

Tuples:

- A collection of ordered, immutable elements enclosed in parentheses ().
- Tuples are immutable, meaning you cannot change their contents.
- Example:

```
my_tuple = (1, "hello", 3.14)
```

Sets:

- An unordered collection of unique elements enclosed in curly braces {}.
- Sets are mutable, meaning you can add or remove elements from them.
- Example:

```
my_set = {1, 2, 3, "hello"}
my_set.add("world")
```

Dictionaries:

- A collection of key-value pairs enclosed in curly braces {}.
- Dictionaries are mutable and unordered.
- Example:

```
person = {"name": "John", "age": 30, "city": "New York"}
person["country"] = "USA"
```

Stacks:

- A collection of elements that supports two main operations: push and pop.
- The last element added to the stack is the first one to be removed (LIFO).
- Example:

```
my_stack = []
my_stack.append(1)
my_stack.append(2)
my_stack.pop()
```

Queues:

- A collection of elements that supports two main operations: enqueue and dequeue.
- The first element added to the queue is the first one to be removed (FIFO).
- Example:

```
from collections import deque
my_queue = deque()
my_queue.append(1)
my_queue.append(2)
my_queue.popleft()
```

Difference between List,Tuples & Set

List:

- Lists are mutable, meaning you can change their contents.
- Lists are ordered, meaning the elements in a list are stored in a specific order.
- Lists can contain duplicate elements.
- Lists are defined using square brackets [].

Tuple:

- Tuples are immutable, meaning you cannot change their contents.
- Tuples are ordered, meaning the elements in a tuple are stored in a specific order.
- Tuples can contain duplicate elements.
- Tuples are defined using parentheses ().

Set:

- Sets are mutable, meaning you can add or remove elements from them.
- Sets are unordered, meaning the elements in a set are not stored in a specific order.
- Sets can only contain unique elements.
- Sets are defined using curly braces {} or the set() function.

Example: Dictionary

Use Dictionary methods to print your favourite tool just by using the keys of the Dictionary

```
#!/usr/bin/python3

fav_tools = {1:"Linux", 2:"Git", 3:"Docker", 4:"Kubernetes",
5:"Terraform", 6:"Ansible", 7:"Chef"}

# Print the value associated with key 1
print(fav_tools[1])

# Alternatively, we can use the get() method to retrieve the value
associated with key 1
print(fav_tools.get(1))
```

Example: List

Create a List of cloud service providers, Write a program to add Digital Ocean to the list of cloud_providers and sort the list in alphabetical order.

```
#!/usr/bin/python3

cloud_providers = ["AWS", "GCP", "Azure"]

cloud_providers.append("Digital Ocean")
cloud_providers.sort()

print(cloud_providers)
```