

"PYTHON'S FANTASTIC FOUR: LISTS, SETS, TUPLES & DICTIONARIES"

INTRODUCTION



In Python, we have four key data structures that help us organize and manage information efficiently.

Think of them like different ways to store and access items:

- 1) Lists – Ordered and changeable, like a to-do list.
- 2) Sets – Unique and unordered, like a bag of different-colored marbles.
- 3) Tuples – Fixed and reliable, like a sealed envelope.
- 4) Dictionaries – Key-value pairs, like a contact list.

●●● LISTS

python

```
fruits = ["apple", "banana", "cherry"]  
print(fruits[0]) # Output: apple
```

Key Features:

- ✓ Ordered – Items stay in a fixed sequence.
- ✓ Mutable – You can add, remove, or change items.
- ✓ Duplicates Allowed – The same item can appear multiple times.

python

```
fruits.append("orange") # Add an item  
fruits.remove("banana") # Remove an item  
fruits.sort()           # Sort the list
```

●●● SETS

python

```
fruits = {"apple", "banana", "cherry"}  
fruits.add("orange")  
print(fruits)  # Order may vary, but no duplicates!
```

Key Features:

- ✓ Unordered – Items don't have a fixed position.
- ✓ Immutable Elements – You can't change items, but you can add/remove them.
- ✓ No Duplicates – Each value appears only once.

python

```
fruits.add("banana")    # No effect, since "banana" is already in the set  
fruits.remove("apple")  # Remove an item  
print(len(fruits))      # Get the number of items
```

●●● TUPLES

python

```
fruits = ("apple", "banana", "cherry")  
print(fruits[0]) # Output: apple
```

Key Features:

- ✓ Ordered – Items stay in a fixed sequence.
- ✓ Immutable – Cannot be changed after creation.
- ✓ Allows Duplicates – The same item can appear multiple times.

python

```
print(len(fruits))      # Get the number of items  
print(fruits[-1])      # Access the last item  
new_tuple = fruits + ("orange",) # Create a new tuple
```

●●● DICT

python

```
student = {"name": "Alice", "age": 20, "grade": "A"}  
print(student["name"]) # Output: Alice
```

- ✓ Key-Value Pairs – Each key maps to a specific value.
- ✓ Unordered – Items don't have a fixed position.
- ✓ Mutable – You can add, remove, or change values.
- ✓ Fast Lookups – Retrieve values quickly using keys.

python

```
student["age"] = 21 # Update a value  
student["city"] = "Paris" # Add a new key-value pair  
del student["grade"] # Remove a key-value pair  
print(student.keys()) # Get all keys
```

LIST



DETAI'NEE

- Список(List) в Python — позволяет хранить различные типы данных, включая числа, строки и даже другие списки, что делает его мощным инструментом для организации и управления информацией. Списки сохраняют порядок элементов. Они также изменяемые, что означает, что вы можете добавлять, удалять или изменять их содержимое. Кроме того, списки допускают дубликаты, поэтому один и тот же элемент может встречаться несколько раз. Благодаря своей гибкости, списки широко используются для работы с пользовательскими данными, хранения записей и управления динамическими наборами данных в программах на Python. 🚀

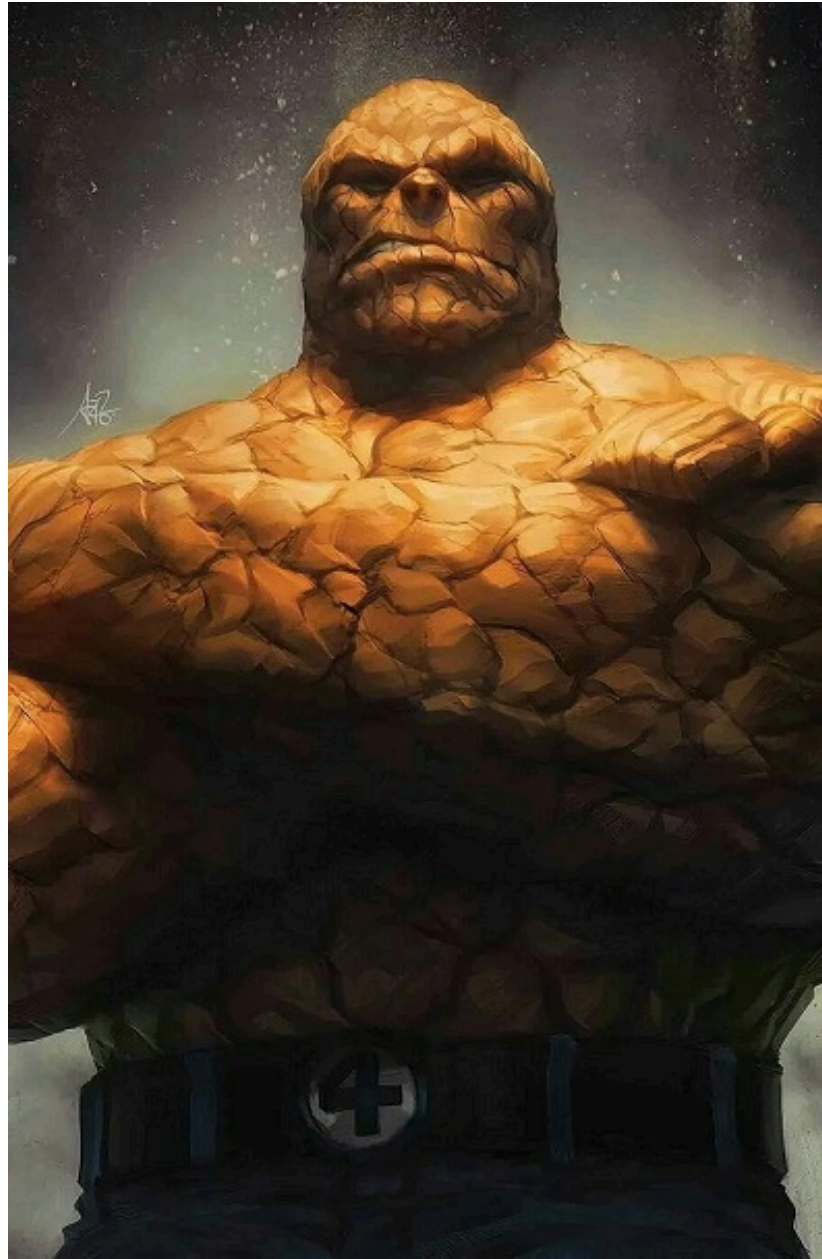
SETS



DETAIL'NEE

- Множество(Set) в Python — это неупорядоченная коллекция уникальных элементов. Множества не допускают дубликатов, что делает их полезными для хранения уникальных значений, таких как идентификаторы пользователей или набор ключевых слов. Порядок элементов в множестве не фиксирован, и при каждом выводе порядок может изменяться.
- Элементы множества неизменяемы, но само множество изменяемое — в него можно добавлять или удалять элементы. Благодаря этим свойствам множества идеально подходят для задач, связанных с обработкой данных, удалением дубликатов и выполнением операций теории множеств, таких как объединение, пересечение и разность. 🚀

TUPLES



DETAIL'NEE

- Кортеж (Tuple) в Python — это упорядоченная, но неизменяемая коллекция элементов. Он работает аналогично списку, но с одним важным отличием: после создания кортеж нельзя изменить — его элементы остаются неизменными. Это делает кортежи идеальными для хранения данных, которые не должны случайно изменяться, например, координат (x, y), дат, дней недели или настроек конфигурации.
- Кортежи поддерживают дубликаты, поскольку порядок элементов фиксирован. Они также занимают меньше памяти и работают быстрее, чем списки, что делает их полезными для работы с большими объемами неизменяемых данных. 🚀

DICTIONARY



DETAIL'NEE

- Словарь(Dictionary) в Python — каждый ключ уникален и связан с определённым значением, что делает словари идеальными для быстрого поиска и структурированного хранения данных.
- В отличие от списков и кортежей, элементы в словаре доступны по ключу, а не по индексу. Словари изменяемы, поэтому можно добавлять, удалять и изменять значения по ключу. Они широко используются для хранения информации, такой как профили пользователей, настройки конфигурации или соответствие между элементами, например, кодами стран и их названиями.
- Благодаря своей гибкости и скорости словари — один из самых мощных инструментов в Python для работы с данными! 🚀

Q&A





Spasibo

