**1. Числовые типы**

* int: Целые числа.
  + Применение: Счетчики, индексы, идентификаторы.
* age = 25

count = 100

* float: Числа с плавающей точкой.
  + Применение: Денежные суммы, измерения, научные вычисления.
* price = 19.99

temperature = 36.6

* complex: Комплексные числа.
  + Применение: Научные вычисления, математические модели.

z = 2 + 3j

**2. Последовательности**

* str: Строки.
  + Применение: Текст, сообщения, имена, адреса.
* name = "Alice"

message = "Hello, World!"

* list: Списки.
  + Применение: Коллекции элементов, которые могут изменяться.
* numbers = [1, 2, 3, 4, 5]

fruits = ["apple", "banana", "cherry"]

* tuple: Кортежи.
  + Применение: Коллекции элементов, которые не должны изменяться.
* coordinates = (10.0, 20.0)

rgb = (255, 0, 0)

* range: Диапазоны.
  + Применение: Генерация последовательностей чисел.
* for i in range(10):

print(i)

**3. Отображения**

* dict: Словари.
  + Применение: Хранение пар "ключ-значение", быстрый доступ по ключу.

person = {"name": "Alice", "age": 25}

**4. Множества**

* set: Множества.
  + Применение: Уникальные элементы, операции над множествами (объединение, пересечение).

unique\_numbers = {1, 2, 3, 4, 5}

* frozenset: Неизменяемые множества.
  + Применение: Уникальные элементы, которые не должны изменяться.

frozen\_set = frozenset([1, 2, 3])

**5. Булевы значения**

* bool: Логические значения.
  + Применение: Условия, флаги, логические операции.
* is\_active = True

has\_permission = False

**6. Байтовые типы**

* bytes: Неизменяемые последовательности байтов.
  + Применение: Работа с бинарными данными, сетевые протоколы.

data = b"hello"

* bytearray: Изменяемые последовательности байтов.
  + Применение: Работа с бинарными данными, которые могут изменяться.

mutable\_data = bytearray(b"hello")

**7. NoneType**

* None: Отсутствие значения.
  + Применение: Инициализация переменных, возврат значений по умолчанию.

result = None

**Примеры применения**

* int: Используйте для счетчиков, индексов и идентификаторов.
* float: Используйте для денежных сумм, измерений и научных вычислений.
* str: Используйте для текста, сообщений, имен и адресов.
* list: Используйте для коллекций элементов, которые могут изменяться.
* tuple: Используйте для коллекций элементов, которые не должны изменяться.
* dict: Используйте для хранения пар "ключ-значение" и быстрого доступа по ключу.
* set: Используйте для уникальных элементов и операций над множествами.
* bool: Используйте для условий, флагов и логических операций.
* bytes: Используйте для работы с бинарными данными и сетевыми протоколами.
* None: Используйте для инициализации переменных и возврата значений по умолчанию.

**1. Списки (Lists)**

Списки — это упорядоченные коллекции элементов, которые могут быть изменены. Элементы списка могут быть любого типа и могут повторяться.

**Создание списка:**

numbers = [1, 2, 3, 4, 5]

fruits = ["apple", "banana", "cherry"]

**Основные операции:**

* Доступ по индексу:

first\_fruit = fruits[0] # "apple"

* Изменение элемента:

fruits[1] = "blueberry"

* Добавление элемента:

fruits.append("date")

* Удаление элемента:

fruits.remove("cherry")

* Длина списка:

length = len(fruits)

* Срезы:

sublist = fruits[1:3] # ["blueberry", "date"]

**Применение:**

* Хранение коллекций элементов, которые могут изменяться.
* Использование в циклах и итерациях.
* Хранение данных, которые могут быть упорядочены.

**2. Кортежи (Tuples)**

Кортежи — это упорядоченные коллекции элементов, которые не могут быть изменены после их создания. Элементы кортежа могут быть любого типа и могут повторяться.

**Создание кортежа:**

coordinates = (10.0, 20.0)

rgb = (255, 0, 0)

**Основные операции:**

* Доступ по индексу:

x = coordinates[0] # 10.0

* Длина кортежа:

length = len(rgb)

* Срезы:

subtuple = rgb[1:3] # (0, 0)

**Применение:**

* Хранение коллекций элементов, которые не должны изменяться.
* Возврат нескольких значений из функции.
* Использование в качестве ключей в словарях (если все элементы кортежа являются неизменяемыми).

**3. Множества (Sets)**

Множества — это неупорядоченные коллекции уникальных элементов. Элементы множества могут быть любого неизменяемого типа.

**Создание множества:**

unique\_numbers = {1, 2, 3, 4, 5}

**Основные операции:**

* Добавление элемента:

unique\_numbers.add(6)

* Удаление элемента:

unique\_numbers.remove(3)

* Объединение множеств:
* set1 = {1, 2, 3}
* set2 = {3, 4, 5}

union\_set = set1 | set2 # {1, 2, 3, 4, 5}

* Пересечение множеств:

intersection\_set = set1 & set2 # {3}

* Разность множеств:

difference\_set = set1 - set2 # {1, 2}

**Применение:**

* Удаление дубликатов из коллекции.
* Проверка наличия элемента.
* Выполнение операций над множествами (объединение, пересечение, разность).

**4. Словари (Dictionaries)**

Словари — это неупорядоченные коллекции пар "ключ-значение". Ключи должны быть уникальными и неизменяемыми, а значения могут быть любого типа.

**Создание словаря:**

person = {"name": "Alice", "age": 25}

**Основные операции:**

* Доступ по ключу:

name = person["name"] # "Alice"

* Добавление или изменение элемента:

person["city"] = "New York"

* Удаление элемента:

del person["age"]

* Проверка наличия ключа:
* if "name" in person:

print("Name exists")

* Получение всех ключей:

keys = person.keys()

* Получение всех значений:

values = person.values()

* Получение всех пар "ключ-значение":

items = person.items()

**Применение:**

* Хранение данных, которые могут быть быстро доступны по ключу.
* Использование в качестве базы данных в памяти.
* Хранение конфигурационных параметров.

**Примеры использования**

* Списки: Хранение списка задач, списка студентов, списка продуктов.
* Кортежи: Хранение координат, RGB значений, возврат нескольких значений из функции.
* Множества: Удаление дубликатов из списка, проверка наличия элемента, выполнение операций над множествами.
* Словари: Хранение информации о пользователях, конфигурационных параметров, данных, которые могут быть быстро доступны по ключу.

Эти типы данных предоставляют мощные инструменты для работы с коллекциями данных в Python и могут быть использованы в различных сценариях в зависимости от ваших потребностей.

/opt/160924-ptm/igor/17\_M/2