

## 2 ЛЕКЦИЯ: ОСНОВНЫЕ ТИПЫ ДАННЫХ В PYTHON

Python — это язык программирования, который поддерживает различные типы данных. Понимание этих типов данных и их особенностей является основой для эффективного программирования. В этой лекции мы рассмотрим основные типы данных: строки, числа, списки, словари и множества.

### 1. Строки (str)

Строки — это последовательности символов, заключенные в одинарные или двойные кавычки.

- Создание:

```
my_string = "Hello, World!"
```

- Методы:

- `.upper()`, `.lower()`, `.replace()`, `.split()`

- Индексация: Строки индексируются, начиная с нуля.

```
print(my_string[0])
```

### 2. Числа (int, float)

Python поддерживает два основных числовых типа: целые числа (int) и числа с плавающей запятой (float).

- Целые числа:

```
my_int = 42
```

- Числа с плавающей запятой:

```
```python
```

```
my_float = 3.14
```

- Операции: Вы можете выполнять арифметические операции: `+`, `-`, `*`, `/`, `//` (целочисленное деление), и `%` (остаток от деления).

### 3. Списки (list)

Списки — это упорядоченные изменяемые коллекции элементов.

- Создание:

```
my_list = [1, 2, 3, 'four']
```

- Методы:

- `.append()`, `.remove()`, `.pop()`, `.sort()`

- Индексация и срезы:

```
print(my_list[1])    # 2
```

```
print(my_list[1:3])  # [2, 3]
```

### 4. Словари (dict)

Словари — это неупорядоченные коллекции пар "ключ-значение".

- Создание:

```
my_dict = {'name': 'Alice', 'age': 25}
```

- Доступ к элементам:

```
print(my_dict['name']) # Alice
```

- Методы:

- `.keys()`, `.values()`, `.items()`, `.get()`

## 5. Множества (set)

Множества — это неупорядоченные коллекции уникальных элементов.

-Создание:

```
my_set = {1, 2, 3, 4}
```

-Методы:

- `.add()`, `.remove()`, `.union()`, `.intersection()`

-Проверка на уникальность: Множества автоматически удаляют дубликаты.

### Заключение

Понимание различных типов данных в Python — это ключ к написанию эффективного и понятного кода. Эти типы данных имеют свои особенности и методы, которые помогают решать различные задачи. Освоив их, вы сможете создавать более сложные структуры данных и программы.