

The background of the slide is a light gray gradient. It is decorated with numerous realistic water droplets of various sizes. Some droplets are large and prominent, while others are small and subtle. They are scattered across the slide, with a higher concentration in the top-left and bottom-right corners, and a few smaller ones near the center text.

QA WITH PYTHON

СТРУКТУРЫ ДАННЫХ.

ЛИСТЫ, КОРТЕЖИ, СЛОВАРИ И МНОЖЕСТВА

СТРУКТУРЫ ДАННЫХ

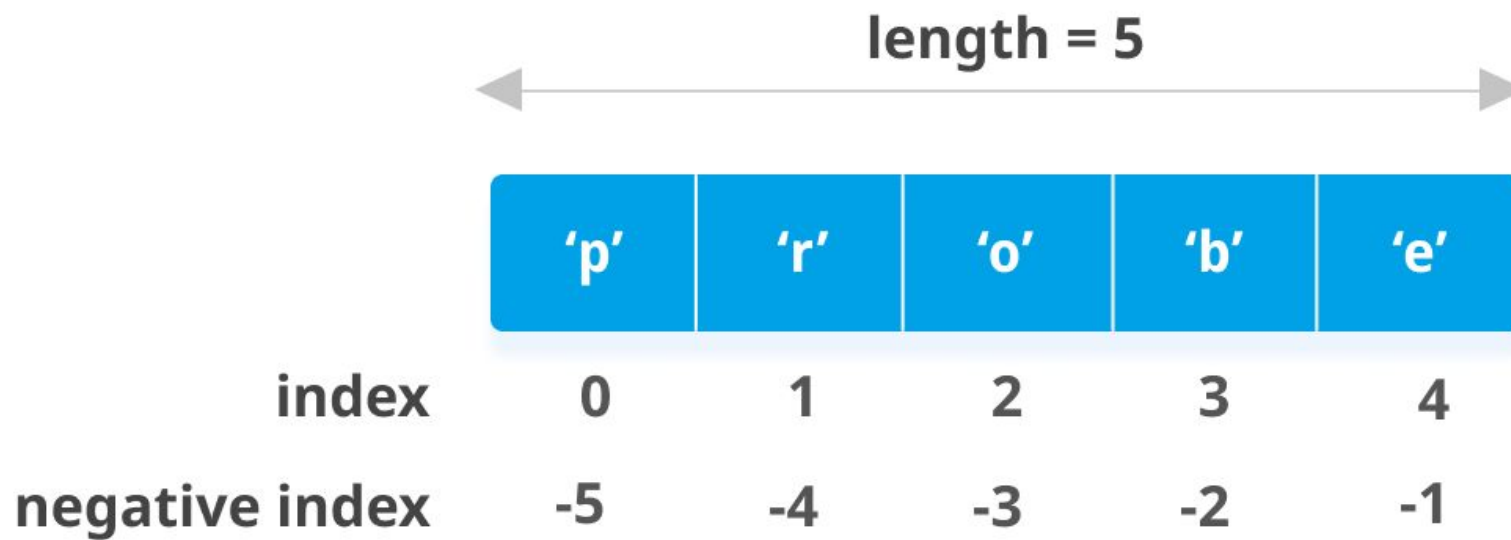
- Списки (lists)
- Кортежи (tuples)
- Словари (dictionaries)
- Множества (sets)



СПИСКИ (LISTS)

- ПРЕДСТАВЛЯЮТ ТИП ДАННЫХ, КОТОРЫЕ ХРАНЯТ НАБОР ИЛИ ПОСЛЕДОВАТЕЛЬНОСТЬ ЭЛЕМЕНТОВ. НАПРИМЕР:
 - МОДЕЛИ АВТОМОБИЛЕЙ;
 - ИМЕНА СТУДЕНТОВ;
 - ПОСЕЩЕННЫЕ СТРАНЫ;
 - ПОСЕТИТЕЛИ МАГАЗИНА И ТАК ДАЛЕЕ;
- МОГУТ ХРАНИТЬ РАЗНЫЕ ТИПЫ ДАННЫХ
- СОЗДАЮТСЯ ПРИ ПОМОЩИ:
 - КВАДРАТНЫХ СКОБОК `MY_LIST = []`
 - ФУНКЦИИ `LIST()` `MY_LIST = LIST()`
- ИЗМЕНЯЕМАЯ СТРУКТУРА ДАННЫХ

ИНДЕКСАЦИЯ



The diagram illustrates string indexing for the word "prebo". A horizontal double-headed arrow at the top is labeled "length = 5". Below it, the characters 'p', 'r', 'o', 'b', and 'e' are each enclosed in a blue square. Underneath these squares, two rows of indices are provided: "index" (0 to 4) and "negative index" (-5 to -1).

	length = 5				
	'p'	'r'	'o'	'b'	'e'
index	0	1	2	3	4
negative index	-5	-4	-3	-2	-1

МЕТОДЫ РАБОТЫ СО СПИСКАМИ

APPEND() МЕТОД ДЛЯ ДОБАВЛЕНИЯ ЭЛЕМЕНТОВ В СПИСОК

.INSERT() ДЛЯ ДОБАВЛЕНИЯ ЭЛЕМЕНТОВ В КОНКРЕТНОЕ МЕСТО В СПИСКЕ

.INDEX() ДЛЯ ПОЛУЧЕНИЯ ИНДЕКСА ЭЛЕМЕНТА

.CLEAR() ДЛЯ ОЧИСТКИ СПИСКА

.REMOVE() ДЛЯ УДАЛЕНИЯ ЭЛЕМЕНТА СПИСКА

.REVERSE() ЧТОБЫ РАЗВЕРНУТЬ СПИСОК В ОБРАТНОМ ПОРЯДКЕ

.COUNT() ДЛЯ ПОДСЧЕТА КОЛИЧЕСТВА ЭЛЕМЕНТОВ В СПИСКЕ

SUM() ДЛЯ СЛОЖЕНИЯ ЭЛЕМЕНТОВ СПИСКА

MIN() ПОКАЗЫВАЕТ ЭЛЕМЕНТ С САМЫМ НИЗКИМ ЗНАЧЕНИЕМ В СПИСКЕ

MAX() ЭЛЕМЕНТ С САМЫМ ВЫСОКИМ ЗНАЧЕНИЕМ В СПИСКЕ

ГЕНЕРАЦИЯ СПИСКОВ (LIST COMPREHENSION)

- ПОЗВОЛЯЕТ ИСПОЛЬЗОВАТЬ СОКРАЩЕННЫЙ СИНТАКСИС ПРИ СОЗДАНИИ НОВОГО СПИСКА НА ОСНОВЕ ЗНАЧЕНИЙ ИЗ УЖЕ СУЩЕСТВУЮЩЕГО СПИСКА
 - `NEWLIST = [EXPRESSION FOR ITEM IN ITERABLE IF CONDITION == TRUE]`

For loop	List Comprehension
<pre>l = [1, 2, 3, 4, 5] new_l = [] for x in l: if x%2: new_l.append(x*x) print(new_l)</pre>	<pre>l = [1, 2, 3, 4, 5] new_l = [x*x for l if x%2] print(new_l)</pre>

КОРТЕЖИ (TUPLES)

- КОРТЕЖИ СОЗДАЮТСЯ С ПОМОЩЬЮ:

- КРУГЛЫХ СКОБОК: (); `MY_TUPLE = ()`

- ФУНКЦИИ `TUPLE()` `MY_TUPLE = TUPLE()`

- ПРИ ПЕРЕЧИСЛЕНИИ НЕСКОЛЬКИХ ЗНАЧЕНИЙ ЧЕРЕЗ ЗАПЯТУЮ `MY_TUPLE = 1, 2, 3`

- ЭЛЕМЕНТЫ ВНУТРИ КОРТЕЖЕЙ РАЗДЕЛЯЮТСЯ ЗАПЯТЫМИ;

- МОГУТ ХРАНИТЬ РАЗНЫЕ ТИПЫ ДАННЫХ

- ВАЖНО СОБЛЮДАТЬ ОСОБЕННОСТИ СИНТАКСИСА, ХАРАКТЕРНЫЕ ДЛЯ КАЖДОГО ОТДЕЛЬНОГО ТИПА ДАННЫХ В КОРТЕЖЕ — КАВЫЧКИ ДЛЯ СТРОК, ЧИСЛА И БУЛЕВЫЕ ЗНАЧЕНИЯ БЕЗ КАВЫЧЕК И ТАК ДАЛЕЕ.

`MY_TUPLE = (1, TRUE, 'NAME', NONE)`

- КОРТЕЖИ – **НЕИЗМЕНЯЕМАЯ** СТРУКТУРА ДАННЫХ

ИСПОЛЬЗОВАНИЕ КОРТЕЖЕЙ

- КОРТЕЖИ ИСПОЛЬЗУЮТСЯ В СЛУЧАЯХ, КОГДА ВАЖНО СОБЛЮДАТЬ ЦЕЛОСТНОСТЬ ДАННЫХ, НАПРИМЕР, КООРДИНАТЫ, НАЗВАНИЯ ШТАТОВ, НАЗВАНИЕ МЕСЯЦЕВ ГОДА И ДНЕЙ НЕДЕЛИ
- С ПОМОЩЬЮ КОРТЕЖЕЙ МОЖНО ПЕРЕДАВАТЬ В ФУНКЦИЮ ПРОИЗВОЛЬНОЕ КОЛИЧЕСТВО АРГУМЕНТОВ – DEF FUNC(*ARGS). СИМВОЛ * ОЗНАЧАЕТ, ЧТО КОРТЕЖ НУЖНО РАСПАКОВАТЬ

```
DEF FUNC(*ARGS):
```

```
    FOR ITEM IN ARGS:
```

```
        RETURN ITEM*ITEM
```

```
PRINT(FUNC(2, 5, 6, 8, 10))
```

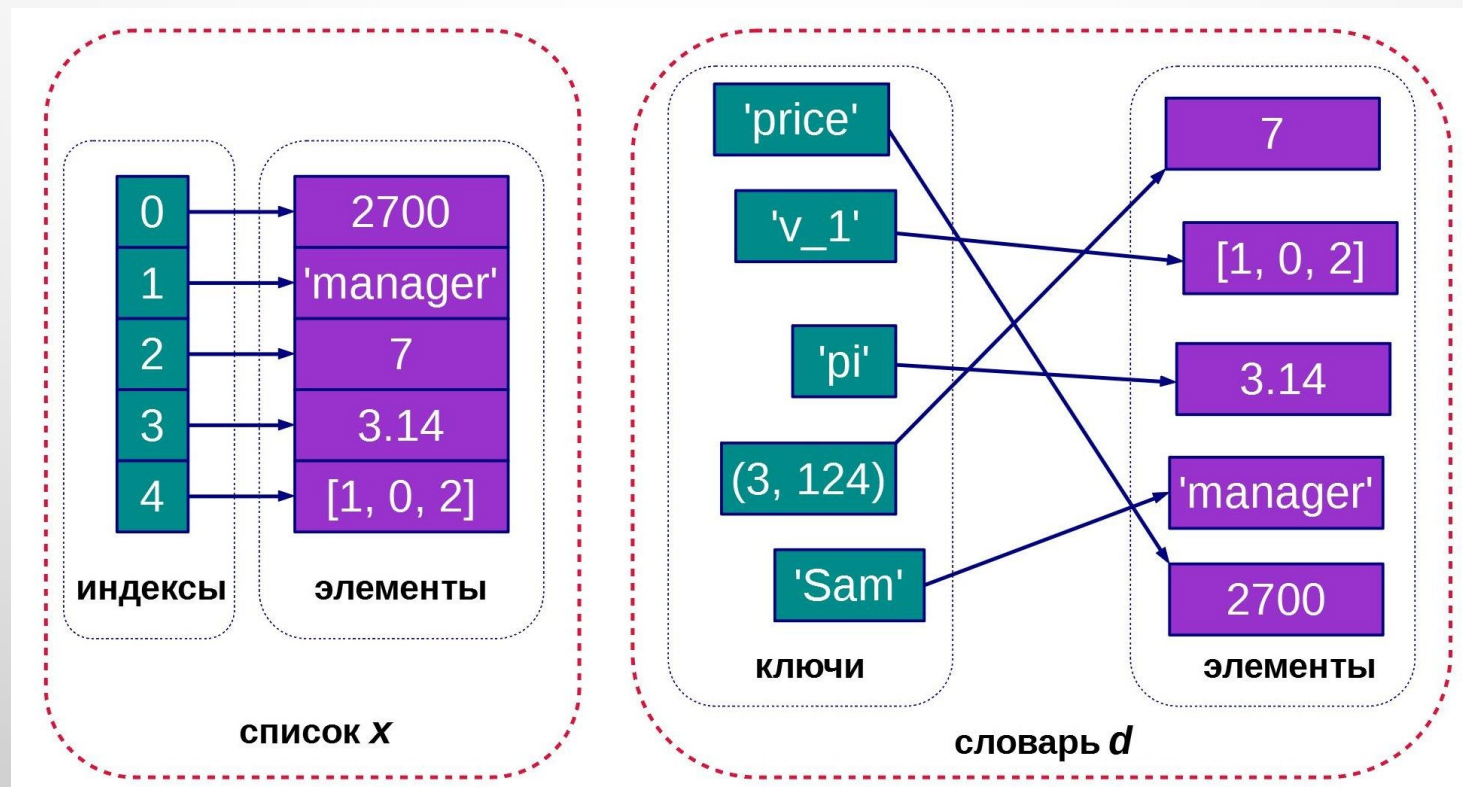

МЕТОДЫ РАБОТЫ С КОРТЕЖАМИ

- `.INDEX()` — ИСПОЛЬЗУЕТСЯ ДЛЯ ВЫВОДА ИНДЕКСА ЭЛЕМЕНТА.
- `.COUNT()` — ИСПОЛЬЗУЕТСЯ ДЛЯ ПОДСЧЕТА КОЛИЧЕСТВА ЭЛЕМЕНТОВ В КОРТЕЖЕ.
- `SUM()` — СКЛАДЫВАЕТ ВСЕ ЭЛЕМЕНТЫ КОРТЕЖА.
- `MIN()` — ПОКАЗЫВАЕТ ЭЛЕМЕНТ КОРТЕЖА С НАИМЕНЬШИМ ЗНАЧЕНИЕМ.
- `MAX()` — ПОКАЗЫВАЕТ ЭЛЕМЕНТ КОРТЕЖА С МАКСИМАЛЬНЫМ ЗНАЧЕНИЕМ.
- `LEN()` — ПОКАЗЫВАЕТ КОЛИЧЕСТВО ЭЛЕМЕНТОВ КОРТЕЖА.

СЛОВАРЬ (DICTIONARY)

- **СЛОВАРЬ** В PYTHON ХРАНИТ КОЛЛЕКЦИЮ ЭЛЕМЕНТОВ, ГДЕ КАЖДЫЙ ЭЛЕМЕНТ ИМЕЕТ УНИКАЛЬНЫЙ КЛЮЧ И СВЯЗАННОЕ С НИМ НЕКОТОРОЕ ЗНАЧЕНИЕ
- СОЗДАЕТСЯ ПРИ ПОМОЩИ:
 - ФИГУРНЫХ СКОБОК `MY_DICT = {}`
 - ФУНКЦИИ `DICT()` `MY_DICT = DICT()`
- ИМЕЕТ СЛЕДУЮЩИЙ СИНТАКСИС:
`DICTIONARY = { КЛЮЧ1:ЗНАЧЕНИЕ1, КЛЮЧ2:ЗНАЧЕНИЕ2,}`
- КЛЮЧИ В СЛОВАРЕ МОГУТ БЫТЬ ТОЛЬКО СТРОКАМИ, ЦЕЛЫМИ ЧИСЛАМИ ИЛИ ЧИСЛАМИ С ПЛАВАЮЩЕЙ ТОЧКОЙ. А ВОТ ЗНАЧЕНИЯ МОГУТ БЫТЬ ЛЮБОГО ТИПА
- ВАЖНО НЕ ЗАБЫВАТЬ ИСПОЛЬЗОВАТЬ КАВЫЧКИ ДЛЯ КЛЮЧА ТИПА `STR`
- ИЗМЕНЯЕМАЯ СТРУКТУРА ДАННЫХ

ОБРАЩЕНИЕ К ЭЛЕМЕНТУ СЛОВАРЯ



МЕТОДЫ РАБОТЫ СО СЛОВАРЯМИ

.KEYS() — ИСПОЛЬЗУЕТСЯ ДЛЯ ВЫВОДА КЛЮЧЕЙ СЛОВАРЯ.

.ITEMS() — ИСПОЛЬЗУЕТСЯ ДЛЯ СОЗДАНИЯ КОРТЕЖЕЙ С КЛЮЧАМИ И ЗНАЧЕНИЯМИ.

.GET() — МЕТОД ДЛЯ ПОЛУЧЕНИЯ ЗНАЧЕНИЯ ПО КЛЮЧУ.

.CLEAR() — ОЧИСТИТЬ СЛОВАРЬ.

.COPY() — СКОПИРОВАТЬ ВСЕ СЛОВАРЬ.

LEN() — ПОЛУЧИТЬ ДЛИНУ СЛОВАРЯ.

TYPE() — УЗНАТЬ ТИП.

MIN() — ПОЛУЧИТЬ КЛЮЧ С МИНИМАЛЬНЫМ ЗНАЧЕНИЕМ.

MAX() — ПОЛУЧИТЬ КЛЮЧ С МАКСИМАЛЬНЫМ ЗНАЧЕНИЕМ.

МНОЖЕСТВА (SETS)

- МНОЖЕСТВО ПОХОЖЕ НА СЛОВАРЬ, В КОТОРОМ ЗНАЧЕНИЯ ОТБРОШЕНЫ, А ОСТАВЛЕНЫ ТОЛЬКО КЛЮЧИ. КАК И В СЛОВАРЕ, КЛЮЧИ ДОЛЖНЫ БЫТЬ УНИКАЛЬНЫМИ. ТО ЕСТЬ ЗНАЧЕНИЯ ВО МНОЖЕСТВЕ НЕ ДОЛЖНЫ ПОВТОРЯТЬСЯ
- СОЗДАЕТСЯ ПРИ ПОМОЩИ:
 - ПЕРЕЧИСЛЕНИЯ ЭЛЕМЕНТОВ В ФИГУРНЫХ СКОБКАХ `MY_SET = {'A','B','C'}`
 - ФУНКЦИИ `SET{}` `MY_SET = SET{}`
- МНОЖЕСТВА УДОБНО ИСПОЛЬЗОВАТЬ ДЛЯ УДАЛЕНИЯ ПОВТОРЯЮЩИХСЯ ЭЛЕМЕНТОВ. К ПРИМЕРУ, ПРИ ВЫПОЛНЕНИИ `SET([1, 8, 2, 1, 5, 8, 9])` ВСЕ ДУБЛИКАТЫ ИСЧЕЗНУТ.

МЕТОДЫ РАБОТЫ СО МНОЖЕСТВАМИ

- `LEN(S)` - ЧИСЛО ЭЛЕМЕНТОВ В МНОЖЕСТВЕ (РАЗМЕР МНОЖЕСТВА).
- `X IN S` - ПРИНАДЛЕЖИТ ЛИ `X` МНОЖЕСТВУ `S`.
- `SET.ADD()` - ДОБАВЛЯЕТ ЭЛЕМЕНТ В МНОЖЕСТВО.
- `SET.REMOVE()` - УДАЛЯЕТ ЭЛЕМЕНТ ИЗ МНОЖЕСТВА. `KEYERROR`, ЕСЛИ ТАКОГО ЭЛЕМЕНТА НЕ СУЩЕСТВУЕТ.
- `SET.POP()` - УДАЛЯЕТ ПЕРВЫЙ ЭЛЕМЕНТ ИЗ МНОЖЕСТВА. ТАК КАК МНОЖЕСТВА НЕ УПОРЯДОЧЕНЫ, НЕЛЬЗЯ ТОЧНО СКАЗАТЬ, КАКОЙ ЭЛЕМЕНТ БУДЕТ ПЕРВЫМ.
- `SET.CLEAR()` - ОЧИСТКА МНОЖЕСТВА
- `SET.ISSUBSET(OTHER)` ИЛИ `SET <= OTHER` - ВСЕ ЭЛЕМЕНТЫ `SET` ПРИНАДЛЕЖАТ `OTHER`.
- `SET.ISSUPERSET(OTHER)` ИЛИ `SET >= OTHER` - ВСЕ ЭЛЕМЕНТЫ `OTHER` ПРИНАДЛЕЖАТ `SET`

ПОЛЕЗНЫЕ ССЫЛКИ

- [HTTPS://WWW.W3SCHOOLS.COM/PYTHON/PYTHON_LISTS.ASP](https://www.w3schools.com/python/python_lists.asp) - LISTS
- [HTTPS://WWW.W3SCHOOLS.COM/PYTHON/PYTHON_TUPLES.ASP](https://www.w3schools.com/python/python_tuples.asp) - TUPLES
- [HTTPS://WWW.W3SCHOOLS.COM/PYTHON/PYTHON_SETS.ASP](https://www.w3schools.com/python/python_sets.asp) - SETS
- [HTTPS://WWW.W3SCHOOLS.COM/PYTHON/PYTHON_DICTIONARIES.ASP](https://www.w3schools.com/python/python_dictionaries.asp) - DICTIONARIES
- [HTTPS://DOCS.PYTHON.ORG/3/LIBRARY/FUNCTIONS.HTML](https://docs.python.org/3/library/functions.html) - PYTHON DOCUMENTATION