

Списки

№ уроку: 6 Курс: Python Starter

Засоби навчання: PyCharm

Огляд, мета та призначення уроку

Після завершення уроку учні матимуть уявлення про списки – структуру даних, яка дозволяє зберігати кілька значень і в Python часто використовується замість масивів в інших мовах програмування, а також основних операцій із цією структурою даних.

Вивчивши матеріал даного заняття, учень зможе:

- Використовувати списки для зберігання та оброблення множини значень
- Отримувати окремі елементи списків та рядків, а також їх перерізи
- Працювати зі списками: додавати та видаляти дані, змінювати існуючі значення, перевіряти елементи на входження до списку
- Розуміти різницю між змінюваними та незмінюваними структурами даних

Зміст уроку

1. Поняття списку
2. Індеси, зрізи списків
3. Перевірка елементів на входження до списку
4. Робота зі списками (отримання значень елементів, їх повторів, додавання та видалення значень, копіювання та видалення списків та їх сортування, перетворення списку на рядок)

Резюме

Масив – набір фіксованої кількості однотипних елементів, розміщених у пам'яті безпосередньо один за одним, доступ до яких здійснюється за індексом (номером).

Індекс – ціле число, яке використовується для доступу до елементів масиву.

Розмірність масиву – максимальна кількість його елементів, чи, інакше, кількість індексів, необхідна для однозначного доступу до елемента масиву.

Масив може мати кілька розмірностей та утворювати двовірні (матриці) або багатовимірні масиви. У більшості мов це реалізовується за допомогою масивів, що складаються з масивів.

Список – це тип даних, що є впорядкованою послідовністю певних значень, які можуть повторюватися.

Кількість елементів списку може бути довільною.

У Python замість масивів зазвичай використовуються списки. Однак у випадках, коли продуктивності списків недостатньо, можна скористатися спеціальними засобами: класом `array` зі стандартної мовної бібліотеки або сторонніми бібліотеками (такими, як `NumPy`). Сторонні рішення пропонують ефективну реалізацію класичних масивів.

Тип даних, що реалізує списки – `List`.

Список у Python є змінюваною (`mutable`) структурою даних. Тобто його елементи можна змінювати після створення списку.

Список у Python є гетерогенною структурою даних. Це означає, що у ньому можна зберігати значення різних типів.

Для створення списку Python слід записати його елементи через кому в квадратних дужках.

Для створення порожнього списку слід просто записати порожні квадратні дужки.

Приклад:

```
my_list = [1, 2, 3]
empty_list = []
```

У Python списки можна індексувати як масиви.

Відлік індексів йде з нуля.

Для отримання елемента за індексом слід після імені списку вказати індекс у квадратних дужках.
Приклад:

```
my_list[2]
```

Також можна індексувати списки з кінця. У такому разі задається негативний індекс, починаючи з -1.

Можна також отримати групу елементів за їх індексами. Ця операція називається зрізом списку (list slicing).

Для отримання зрізу списку в квадратних дужках вказується індекс першого елемента, потім через двокрапку записується індекс останнього необхідного елемента, збільшений на одиницю:

```
my_list[start:end]
```

Можна також додати ще одну двокрапку та крок зрізу:

```
my_list[start:end:step]
```

Можна не вказувати початковий та/або кінцевий індекси. Значенням за замовчуванням для початкового індексу є 0, для кінцевого – довжина списку. Якщо не вказано ні початкового, ні кінцевого індексів, ні кроку, то `my_list[:]` поверне копію списку.

Для перевірки, чи вказане значення є елементом списку, використовується операція `in`.

Для отримання кількості елементів у списку використовується функція `len`.

Для додавання нового елемента до списку використовується метод `append()`:

```
my_list.append(value)
```

Для додавання декількох нових елементів одночасно до списку використовується метод `extend()`:

```
value = value1, value2, value3
```

```
my_list.extend(value)
```

Для видалення елемента списку використовується метод `remove()` – у разі відсутності елемента `value` у списку буде помилка:

```
my_list.remove(value)
```

Для видалення елемента списку за індексом використовується:

оператор `del`: `del my_list[index]`

метод `pop()`: `pop(index)` – (якщо індекс не вказано – видалення останнього елемента з кінця):

```
my_list.pop(index)
```

Ще однією особливістю даного методу є те, що елемент, який видаляється, можна покласти в змінну:

```
my_var = my_list.pop(index)
```

Для копіювання списку використовується метод `copy()`:

```
my_list2 = my_list1.copy()
```

Для вставки елемента в список використовується метод `insert()`:

```
my_list.insert(index, value)
```

Для підрахунку кількості певних значень у списку використовується метод `count()` :

```
my_list.count(value)
```

Для отримання порядкового номера першого елемента зі значенням `value` в діапазоні від `i` до:

```
my_list.index(value, upper_range_index, upper_range_index))
```

Для склеювання елементів списку в рядок використовується метод `join()`:

```
char_list = ["a", "e", "i", "o", "u", "y"]
str_list = ";".join(char_list)
```

Для очищення списку використовується метод `clear()`:
`my_list.clear()`

Для сортування списку за зростанням використовується метод `sort()`, який змінює вихідний:
`my_list.sort()`

Для сортування списку за спаданням використовується метод `sort()`, який змінює вихідний:
`my_list.sort(reverse=True)`

Для сортування списку за зростанням використовується метод `sorted()`, який повертає новий:
`my_list2 = sorted(my_list1)`

Для сортування списку за спаданням використовується метод `sorted()`, який повертає новий:
`my_list2 = sorted(my_list1, reverse=True)`

Для виведення списку в оберненому порядку використовується метод `reverse()`, який змінює вихідний: `my_list.reverse()`

Для виведення списку в оберненому порядку використовується метод `reversed()`, який повертає ітерований об'єкт, тому що побачити список треба залучити функцію `list()`:
`print(list(reversed(my_list)))`

Для перебору елементів списку використовується цикл `for`:
for `element` **in** `my_list`:

```
# ...
# виконання операцій з element
# ...
pass
```

Закріплення матеріалу

- Що таке масив?
- Що таке індекс?
- Що таке список?
- Чи є класичні масиви частиною мови Python? Що використовують замість них?
- Як створити список у Python?
- Який індекс має перший елемент списку?
- Який індекс має останній елемент списку?
- Що таке переріз списку?
- Як дізнатися довжину списку?
- Як додати новий елемент до списку?
- Як видалити елемент зі списку?
- Як скопіювати список?
- Як видалити список?
- Як дізнатися позицію елемента у списку?
- Як обробити кожний елемент списку?

Додаткове завдання

Завдання 1

Створіть список, введіть кількість його елементів і самі значення. Передбачити меню, в якому: після натискання клавіші 1 ці значення виведуться на екран у зворотному порядку, а після натискання клавіші 2 – за зростанням.

Завдання 2

Створіть список натуральних чисел `int_list`. Кожне непарне значення списку додайте до нового списку `new_list`. Користувач вводить з клавіатури кількість повторів списку `repeat`. Здійсніть дублювання списку `new_list`, `repeat` кількість разів. Очистіть список `int_list`.

Завдання 3

Для цього завдання вихідний список значень беремо з підсумкового списку `new_list` додаткового завдання 2. Користувач вводить з клавіатури значення; якщо таке є у списку — вивести кількість його повторів та його позицію у цьому списку.

Самостійна діяльність учня

Завдання 1

Створіть список та введіть його значення. Знайдіть найбільший та найменший елемент списку, а також суму та середнє арифметичне його значень.

Завдання 2

Перепишіть рішення останнього завдання з шостого уроку так, щоб воно не використовувало рекурсію і не обчислювало всі проміжні кількості варіантів шляхів багато разів (що вкрай неефективно), а зберігало їх у списку.

Завдання 3

Простим називається число, яке ділиться націло лише на одиницю і на саме себе. Число 1 не вважається простим. Напишіть програму, яка знаходить усі прості числа в заданому проміжку, виводить їх на екран, а потім на вимогу користувача виводить їхню суму або добуток.

Рекомендовані ресурси

Документація з Python

<https://docs.python.org/3/library/stdtypes.html#list>

Статті у Вікіпедії про ключові поняття, розглянуті на цьому уроці

[https://uk.wikipedia.org/wiki/Масив_\(структура_даних\)](https://uk.wikipedia.org/wiki/Масив_(структура_даних))

<https://ru.wikipedia.org/wiki/Список>