

# Послідовності

№ уроку: 5 Курс: Python Starter

Засоби навчання: PyCharm

## Огляд, мета та призначення уроку

Після завершення уроку учні матимуть уявлення про послідовності в Python та основні стандартні послідовності, їх призначення та використання, зможуть реалізовувати отримані знання на практиці.

## Вивчивши матеріал даного заняття, учень зможе:

- Мати уявлення про послідовності
- Визначати, які контейнери в Python вважаються послідовностями та створювати свої послідовності.
- Знати основні операції, спільні для всіх послідовностей та змінюваних послідовностей
- Працювати з кортежами
- Працювати з іменованими кортежами

## Зміст уроку

1. Що таке послідовність?
2. Операції над послідовностями
3. Кортежі (tuple)
4. Іменовані кортежі (namedtuple)
5. Перевірка на входження елементу до послідовності

## Резюме

**Послідовність** – це впорядкована колекція, яка підтримує індексований доступ до елементів.

Деякі послідовності в Python на відміну від традиційних масивів (наприклад, в C) можуть зберігати елементи різного типу (у тому числі колекції різних видів). У мові Python є чотири вбудовані типи послідовностей: list (розглянемо докладно на наступному занятті), range, str, bytes, bytearray, tuple.

### Операції над послідовностями

Деякі ітерабельні об'єкти мають певні спільні властивості.

Ітерабельні об'єкти, які підтримують ефективний доступ до елементів з використанням цілих індексів через спеціальний метод `__getitem__()` і підтримують метод `__len__()`, який повертає кількість елементів, називаються послідовностями.

Довжина `len(s)`: функція `len()` повертає довжину (кількість елементів у послідовності) `s`.

Конкатенація (**склеювання**): `s + t` – повертає новий об'єкт-склеюку `s` та `t`. Дублювання `s * n` – повертає послідовність, що повторюється `n` раз.

**Індексація та зрізи.** Отримати доступ до окремого або групи елементів послідовності можна за допомогою оператора `[]`. Індексацію (отримання окремого елементу) можна вважати окремим випадком отримання зрізу (**слайсингу**).

Оператор отримання зрізу має три форми запису:

- `s[start]` – індексація (з 0);
- `s[start:end]` – зріз `[start; end]`;
- `s[start:end:step]` – зріз `[start; end]` з кроком `Step`.

У ряді випадків цілі параметри `start`, `end` і `step` можуть бути опущені. Елемент з індексом `end` **не включається** до результату при взятті зрізів.

Мінімальне та максимальне значення `min(s)` та `max(s)` — повертає мінімальний та максимальний елементи послідовності `s` відповідно. Перевірка на входження `x in s` — повертає `True`, якщо `x` входить у послідовність `s` і `False` в іншому випадку.

**Індекс (положення) елементу** `s.index(x[, start[, end]])` --> `int` — повертає перше входження елементу `x` у послідовність `s` (між індексами `start` та `end`, якщо вони задані).

Кількість повторень `s.count(x)` — повертає кількість входжень елементів `x` до послідовності `s`.

Сортування `sorted(iterable, key = None, reverse = False)` - повертає відсортований об'єкт у вигляді списку. Вихідний об'єкт при цьому не змінюється.

Параметри:

**key** — функція сортування (за замовчуванням не враховується, сортування здійснюється поелементно)

**reverse** — якщо дорівнює `True`, сортування здійснюється у зворотному порядку.

Раніше вже було розглянуто три типи даних, які є послідовностями.

## Кортежі

**Кортеж (tuple)** — це впорядкована **незмінювана** послідовність елементів.

Особливості:

- Вміє все, що вміє список, за винятком операцій, що призводять до зміни кортежу.
- Застосовується у випадках, коли відомо, що послідовність не змінюватиметься після створення.

## Перевірка входження елементу

Більшість послідовностей підтримують операції перевірки входження елементу `in` та `not in`:

```
>>> 4 in (1, 5, 4)
True
>>> 40 in range(30, 100)
True
>>> "i" in "ispum"
True
>>> 4 not in (3, 4, 5)
False
>>>
```

## Закріплення матеріалу

- Що таке послідовність?
- Які способи має реалізовувати будь-яка послідовність?
- Які методи реалізують більшість послідовностей?
- Які методи реалізують більшість змінюваних послідовностей?
- Які ви знаєте вбудовані незмінні послідовності в Python?
- Які ви знаєте вбудовані послідовності, що змінюються, в Python?
- Що таке кортежі?
- Що таке іменовані кортежі?
- Як реалізується передача довільної кількості аргументів функції Python?

## Додаткове завдання

### Завдання 1

Напишіть програму, яка вводить з клавіатури послідовність чисел, перетворює послідовність на кортеж і виводить його відсортованим у порядку зростання.

### Завдання 2

Напишіть програму для аналізу фідбеку від відвідувачів курорту «Морська зірка», яка повинна знаходити згадки про меню, спортзал, обслуговування (за кожне знайдене співпадіння нараховується 5% знижки на наступне відвідування). Якщо фідбек перевищив 60 символів, відвідувачу надається додаткова знижка 15% на наступне відвідування.

## Самостійна діяльність учня

### Завдання 1

Створіть програму, яка зчитує рядок, в якому знаходиться ПІБ користувача і перевіряє, чи складається рядок з літер, при чому кожне слово має бути записане з великої літери. Вивести результат на екран.

### Завдання 2

Напишіть програму, в якій користувач вводить із клавіатури діапазон чисел (в діапазоні має бути не менше 5 чисел). Вивести на екран суму другого, передостаннього, а також середнього арифметичного значення даної послідовності.

### Завдання 3

Напишіть програму, яка на вхід отримує параметри кольору (в діапазоні від 0 до 255 для кожного кольору) у форматі RGB і виводить на екран кортеж, у якому зберігається колір.

### Завдання 4

Ознайомтеся за допомогою документації з класами `namedtuple` та `deque` модуля `collections`. Створіть фабрику іменованих кортежів оцінок для учнів однієї групи з предметів: алгебра, геометрія, історія, інформатика, географія. Вивести дані на екран.

## Рекомендовані ресурси

Документація Python

<https://docs.python.org/3/glossary.html#term-sequence>

<https://docs.python.org/3/tutorial/datastructures.html>  
<https://docs.python.org/3/library/stdtypes.html#sequence-types-list-tuple-range>  
<https://docs.python.org/3/library/stdtypes.html#common-sequence-operations>  
<https://docs.python.org/3/library/stdtypes.html#immutable-sequence-types>  
<https://docs.python.org/3/library/stdtypes.html#mutable-sequence-types>  
<https://docs.python.org/3/library/stdtypes.html#lists>  
<https://docs.python.org/3/library/stdtypes.html#tuples>  
<https://docs.python.org/3/library/stdtypes.html#ranges>  
<https://docs.python.org/3/library/stdtypes.html#text-sequence-type-str>  
<https://docs.python.org/3/library/stdtypes.html#string-methods>  
<https://docs.python.org/3/tutorial/controlflow.html#arbitrary-argument-lists>  
<https://docs.python.org/3/tutorial/controlflow.html#unpacking-argument-lists>  
<https://docs.python.org/3/library/collections.html>

Статьи в Википедии о ключевых понятиях, рассмотренных на этом уроке

[https://ru.wikipedia.org/wiki/Список\\_\(информатика\)](https://ru.wikipedia.org/wiki/Список_(информатика))  
[https://ru.wikipedia.org/wiki/Строковый\\_тип](https://ru.wikipedia.org/wiki/Строковый_тип)  
[https://ru.wikipedia.org/wiki/Списковое\\_включение](https://ru.wikipedia.org/wiki/Списковое_включение)