



# Лекція 2. Робота з структурами даних в Python

Лектор – доцент, кандидат фізико-математичних наук  
**Кириченко Віктор Вікторович**



## ПЛАН ЛЕКЦІЇ

1. Основні структури даних
2. Методи і функції для роботи з структурами даних



# Примітивні типи даних та змінні в Python. Неявна типізація. Перетворення типів.

**Список (list) - впорядкована змінна колекція елементів.**

Синтаксис: `my_list = [1, 2, 3]`

Особливості: підтримує зміну елементів, дублікати, індексацію.

Приклад:

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

```
fruits.append("orange")
```



# Примітивні типи даних та змінні в Python. Неявна типізація. Перетворення типів.

**Кортеж (tuple) - впорядкована незмінна колекція.**

Синтаксис: `my_tuple = (1, 2, 3)`

Особливості: швидший за список, можна використовувати як ключ у словнику.

Приклад:

```
point = (10, 20)
```



# Примітивні типи даних та змінні в Python. Неявна типізація. Перетворення типів.

**Множина (set) - неупорядкована колекція унікальних елементів.**

Синтаксис: `my_set = {1, 2, 3}`

Особливості: автоматично уникає дублікатів, підтримує операції над множинами.

Приклад:

```
unique_items = {1, 2, 2, 3}  # результат: {1, 2, 3}
```



# Примітивні типи даних та змінні в Python. Неявна типізація. Перетворення типів.

**Словник (dict) - набір пар "ключ: значення"**

Синтаксис: `my_dict = {"key": "value"}`

Особливості: дуже зручний для асоціативного зберігання.

Приклад:

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

```
person["age"] += 1
```



# Методи і функції для роботи з структурами даних

## Списки (list)

|                           |                                                         |
|---------------------------|---------------------------------------------------------|
| <code>append(x)</code>    | # Додає x в кінець списку                               |
| <code>extend(iter)</code> | # Додає всі елементи з іншої колекції                   |
| <code>insert(i, x)</code> | # Вставляє x на позицію i                               |
| <code>remove(x)</code>    | # Видаляє перше входження x                             |
| <code>pop(i)</code>       | # Видаляє i повертає елемент з позиції i (або останній) |
| <code>index(x)</code>     | # Повертає індекс першого входження x                   |
| <code>count(x)</code>     | # Кількість входжень x                                  |
| <code>sort()</code>       | # Сортує список                                         |
| <code>reverse()</code>    | # Реверсує список                                       |
| <code>clear()</code>      | # Очищає список                                         |



# Методи і функції для роботи з структурами даних

## Кортежі (tuple)

`count(x)`                   # Кількість входжень `x`

`index(x)`                   # Індекс першого входження `x`

Не можна змінювати (немає методів `append`, `remove`, тощо).



# Методи і функції для роботи з структурами даних

## Множини (set)

```
add(x)           # Додає елемент x
remove(x)        # Видаляє x (KeyError, якщо нема)
discard(x)       # Видаляє x, якщо існує
pop()            # Видаляє випадковий елемент
clear()          # Очищає множину
union(s)         # Об'єднання
intersection(s)  # Перетин
difference(s)    # Різниця
symmetric_difference(s) # Симетрична різниця
issubset(s)      # Чи є підмножиною
issuperset(s)    # Чи є надмножиною
```



# Методи і функції для роботи з структурами даних

## Словники (dict)

|                                |                                                   |
|--------------------------------|---------------------------------------------------|
| <code>get(k[,d])</code>        | # Повертає значення за ключем k або d (якщо нема) |
| <code>keys()</code>            | # Список ключів                                   |
| <code>values()</code>          | # Список значень                                  |
| <code>items()</code>           | # Пари (ключ, значення)                           |
| <code>update(d2)</code>        | # Додає пари з d2                                 |
| <code>pop(k[,d])</code>        | # Видаляє ключ і повертає значення                |
| <code>popitem()</code>         | # Випадково (з 3.7 — останній) ключ-значення      |
| <code>setdefault(k[,d])</code> | # Якщо ключа нема — додає його зі значенням d     |
| <code>clear()</code>           | # Очищення                                        |



# Методи і функції для роботи з структурами даних

## Функції для всіх ітерабельних структур

|                              |                                           |
|------------------------------|-------------------------------------------|
| <code>len(data)</code>       | # Кількість елементів                     |
| <code>sorted(data)</code>    | # Повертає відсортований список           |
| <code>sum(data)</code>       | # Сума елементів (якщо можна додати)      |
| <code>min(data)</code>       | # Мінімум                                 |
| <code>max(data)</code>       | # Максимум                                |
| <code>list(iterable)</code>  | # Перетворення в список                   |
| <code>set(iterable)</code>   | # Перетворення в множину                  |
| <code>tuple(iterable)</code> | # Перетворення в кортеж                   |
| <code>enumerate(data)</code> | # Ітерація з індексом                     |
| <code>zip(a, b)</code>       | # Об'єднання кількох колекцій поелементно |
| <code>any(data)</code>       | # True, якщо хоч один елемент істинний    |
| <code>all(data)</code>       | # True, якщо всі елементи істинні         |



**ДЯКУЮ ЗА УВАГУ**