

# Модулі Python

**№ уроку:** 7 **Курс:** Python Базовий

**Засоби навчання:** Персональний комп'ютер/ноутбук стандартної продуктивності

## Огляд, мета та призначення уроку

Ознайомитись з тим, що таке модулі/пакети. На уроці буде розглянуто, як користуватися модулями в завданнях, для чого вони потрібні і які вбудовані модулі вже є в Python 3. Також буде розглянуто як написати власний модуль.

## Вивчивши матеріал даного заняття, учень зможе:

- Розуміти, що таке модуль/пакет і які вбудовані модулі в Python 3.
- Вміти вирішувати задачі застосовуючи вивчені модулі.

## Зміст уроку

1. Які є модулі в Python 3
2. Які готові функції у них є
3. Розв'язання задач із даними модулями

## Резюме

- Модуль (пакет) - це набір готових класів, функцій для вирішення певних завдань.
- Для того, щоб користуватися модулем, його потрібно імпортувати, використовуючи ключове слово `import` і вказати ім'я модуля `module_name`. Таким чином ви імпортуєте відразу весь модуль з усіма його класами та функціями.
- По суті, модуль - це папка, в якій можуть бути інші папки з файлами або файли. Допустимо у модулі є файл `functions.py`, у якому є клас `Function`. Якщо ви хочете імпортувати тільки цей клас, щоб потім створити його об'єкт, використовуйте наступний синтаксис: `from my_module.functions import Function`. Аналогічно з іншими структурами всередині модуля. Якби `functions.py` лежав у модулі в папці методів, імпортувати його треба було б так:  
`from my_module.methods.functions import Function`.
- Якщо вам потрібно зробити кілька імпортів з одного модуля, ви можете просто перерахувати їх через кому::  
`from my_module.functions import function_1, function_2, function_x`
- Також у модулях підтримується `alias` (альтернативне ім'я). Наприклад, якщо ви робите імпорт (будь-який), наприклад:  
`import my_long_module_name_module`, то ви можете скористатися або зробити ось так:  
`import my_long_module_name_module as short_mod`  
чи взагалі:  
`import my_long_module_name_module as m`  
і користуватися вже у коді вибраним вами ім'ям.

- Одні з найважливіших базових модулів: collections (для роботи зі структурами даних), itertools (для роботи з функціями комбінаторики), datetime (робота з датами та часом), random (генерація випадкових значень), math (математичні функції), re (робота з регулярними виразами).
- Щоб написати свій модуль, достатньо створити папку з ім'ям, яке ви хочете дати модулю, а в ній створити порожній файл init.py, після чого всередині ви вже можете створювати будь-яку структуру. Тільки для кожного підмодуля (тобто міні модулі всередині основного, це хороша практика) аналогічно потрібно створювати в папці порожній файл init.py . Цей файл є своєрідним конструктором модуля як і в класах метод init. Тобто він повинен бути і код у ньому (якщо є) буде виконаний як підготовчий скрипт перед імпортом самого модуля або його функцій, змінних чи класів.

### Закріплення матеріалу

- Що таке пакет/модуль?
- Навіщо потрібні модулі?
- Що потрібно зробити, щоб написати власний модуль?
- Які модулі було вивчено на уроці? За що вони відповідають?

### Додаткове завдання

Вивчіть які ще існують вбудовані модулі Python. Навіщо вони потрібні? Які є корисні методи? Знайдіть щонайменше 3.

### Самостійна діяльність учня

Напишіть функцію-калькулятор. Вона має виконувати крім базових операцій (+, -, /, \*) ще такі: зведення у ступінь, квадратний корінь, логарифм з обраної основи. Функція повинна приймати 3 аргументи: a, b, op, де a та b це числа для операцій над ними, а op ім'я операції у вигляді рядка, зрозумілого користувачеві. Для того, щоб використовувати функцію на ім'я, зробіть словник (він може бути як глобальної видимості, так і всередині функції), де ключі - це передбачувані назви функцій, а значення - це функції (без виклику).

Наприклад: f = {"sum\_a\_b": lambda a,b: a+b}, f["sum\_a\_b"](2, 2) буде 4. Аналогічно інших базових операцій . Для операцій корінь та інших, використовуйте тільки ім'я самої функції, наприклад {"sqrt": math.sqrt} якщо ви імпортували весь пакет math або sqrt, якщо ви зробили from math import sqrt

### Рекомендовані ресурси

- [Modules](#)
- [Модулі 1](#)
- [Модулі 2](#)
- [Модулі 3](#)
- [Модулі 4](#)