

# Регулярні вирази

№ уроку: 10 Курс: Python Essential

Засоби навчання: PyCharm

## Огляд, мета та призначення уроку

Після завершення уроку учні матимуть уявлення про регулярні вирази (Regex) – рядки, що задають шаблон для пошуку певних фрагментів у тексті, яким чином їх можна задати, і на практиці використовувати в програмах на Python.

## Вивчивши матеріал даного заняття, учень зможе:

- Розуміти основи парадигми об'єктно-орієнтованого програмування
- Розуміти призначення та різницю між класами та об'єктами
- Створювати класи
- Створювати екземпляри класів
- Створювати методи

## Зміст уроку

1. Що таке регулярні вирази?
2. Регулярні вирази у Python
3. Основні інструменти для роботи з RegEx
4. Основні метасимволи в RegEx
5. Використання в Python

## Резюме

**Регулярні вирази (Regex)** – це рядки, які задають шаблон для пошуку певних фрагментів у тексті. Крім пошуку, за допомогою спеціальних Regex-шаблонів можна маніпулювати текстовими фрагментами – видаляти та змінювати підрядки частково чи повністю.

### Regex в Python

Більшість сучасних високорівневих мов програмування підтримують регулярні вирази, проте ступінь зручності використання Regex у різних мовах варіюється. Python надає прості та зрозумілі методи для роботи з регулярними виразами. Всі інструменти Regex знаходяться в модулі **re**, який входить до стандартної бібліотеки мови Python.

Для того, щоб розпочати роботу з регулярними виразами в Python, необхідно імпортувати модуль **re** у свій проект за допомогою наступної команди:

**import re**

Для екранування службових символів у шаблонах пошуку та заміни використовують два способи: або зворотний слеш `\`, або «сирі» рядки `r''`. Другий метод дозволяє уникнути нагромадження слешів у шаблонах, тому найчастіше перевага надається саме йому.

Розглянемо основні функції модуля **re**:

**re.match(pattern, string)** – знаходить входження фрагмента **на початку рядку**.

**re.search(pattern, string)** – схожа на функцію **match()**, але здійснює пошук заданого шаблону по всьому рядку, повертає лише перше знайдене співпадіння.

**re.findall(pattern, string)** – функція, призначена для пошуку за заданим шаблоном та повернення **всіх знайдених значень**.

**re.finditer(pattern, string)** – функція, використовується для пошуку всіх співпадінь у шаблоні, які не пересікаються.

Повертає ітератор із об'єктами Match.

Функція finditer() повертає ітератор навіть у тому випадку, коли співпадіння не знайдено.

Функція finditer відмінно підходить для обробки тих команд, виведення яких відображається стовпцями.

**re.sub(pattern, repl, string)** – функція, призначена для пошуку за заданим шаблоном та заміни на зазначений підрядок.

**re.split(pattern, string, [maxsplit=0])** – функція, призначена для ділення рядка за заданим шаблоном на таку кількість поділів, яка тільки можлива.

**re.compile(pattern, repl, string)** – можливість зібрати регулярний вираз в об'єкт, який в свою чергу можна використовувати для пошуку. Призначений для пошуку за заданим шаблоном і дозволяє уникнути переписування одного і того ж коду (виразу).

### Основні прапори модуля re

Короткий синтаксис	Повний синтаксис	Призначення
re.A	re.ASCII	Повертає співпадіння лише по ASCII-символам замість усієї таблиці Unicode.
re.I	re.IGNORECASE	Ігнорує регістр символів.
re.M	re.MULTILINE	Використовується спільно з метасимволами ^ та \$. У першому випадку повертає співпадіння на початку кожного нового рядка \n, у другому – наприкінці \n.
re.S	re.DOTALL	Змушує метасимвол . повертати співпадіння за абсолютно всіма символами, включаючи \n. Без цього прапора крапка . відповідає будь-якому символу, крім \n.
re.X	re.VERBOSE	Дозволяє коментарі в Regex-виразах.
re.L	re.LOCALE	Враховує регіональні налаштування при використанні \w, \W, \b, \B, \s i \S. Використовується <b>тільки</b> при роботі з байтовими рядками, несумісний з re.ASCII.

### Основні метасимволи в Regex

Повний синтаксис	Призначення
.	Один будь-який символ, крім нового рядка \n
?	0 або 1 входження шаблону зліва

+	1 і більше входжень шаблону зліва
*	0 і більше входжень шаблону зліва
\w	Будь-яка цифра чи літера
\W	Все, окрім літери або цифри
\d	Будь-яка цифра [0-9]
\D	Все, окрім цифри
\s	Будь-який символ пробілу
\S	Будь-який непробільний символ
\A	Чи починається рядок з певної послідовності символів
\Z	Чи закінчується рядок потрібною послідовністю символів
\b	Повертає співпадіння, якщо слово починається або закінчується потрібною послідовністю символів
\B	Повертає співпадіння, якщо певні символи є у рядку, але не на початку чи не наприкінці слова
[..]	Один із символів у дужках
[^..]	Будь-який символ, крім тих, що у дужках
\	Екранування спеціальних символів (\. означає крапку, або \+ - знак «плюс»)
^	Початок рядку

\$	Кінець рядку
{n,m}	Від n до m входжень ({,m} — від 0 до m)
a b	Відповідає a або b
()	Групує вираз та повертає знайдений текст
\t, \n, \r	Символ табуляції, нового рядка та повернення каретки відповідно

### Онлайн-конструктори регулярних виразів

Чим складніший регулярний вираз, тим важче його правильно скласти та протестувати. В інтернеті є чимало візуалізаторів RegEx, які значно спрощують це завдання. Найзручніший ресурс – [regex101](http://regex101.com). Сайт надає довідкову та налагоджувальну інформацію, дозволяє візуально тестувати шаблони для пошуку та заміни. Крім Python підтримує PHP, Java, Golang та JavaScript.

### Закріплення матеріалу

- Що таке регулярні вирази?
- Який модуль в Python дає можливість використовувати регулярні вирази?
- Які функції входять до його складу?
- Які прапори можна використовувати у процесі формування регулярних виразів?
- Які основні метасимволи можна використовувати у процесі формування регулярних виразів?
- З якою метою використовують онлайн-конструктори RegEx?

### Додаткове завдання

#### Завдання 1

Напишіть функцію, яка буде аналізувати текст, що надходить до неї, і виводити тільки унікальні слова на екран, загальну кількість слів і кількість унікальних слів.

#### Завдання 2

З клавіатури вводиться рядок, в якому є інформація про прізвище, ім'я, дату народження, електронну адресу та відгук про курси учня. Написати функцію, яка, використовуючи регулярні вирази, витягне дані з рядка і поверне словник.

### Самостійна діяльність учня

#### Завдання 1

Написати функцію, яка за допомогою регулярних виразів розбиває текст на окремі слова і знаходить частоту окремих слів.

#### Завдання 2

Написати функцію, яка за допомогою регулярних виразів з файлу витягує дані про дату народження, телефон та електронну адресу. Дані потрібно записати до іншого файлу.

#### Завдання 3

Користувач вводить з клавіатури пропозицію. Написати функцію, яка друкуватиме на екран останні 3 символи кожного слова.

#### Рекомендовані ресурси

[https://uk.wikipedia.org/wiki/Регулярний вираз](https://uk.wikipedia.org/wiki/Регулярний_вираз)

Документація з Python

<https://docs.python.org/3.8/library/re.html>

Онлайн-конструктори регулярних виразів

<https://regex101.com/>