Посилання на репозиторій GitHub

<https://github.com/Danilo340/pythonlesson.git>

**Використані бібліотеки**

Бібліотека *re*: Це стандартна бібліотека Python для роботи з регулярними виразами. В даному коді я використав метод re.match для перевірки, чи відповідає вхідна строка заданому шаблону.

**Логіка коду**

Для початку імпортую бібліотеку re. import re

Після чого роблю функцію для перевірки валідації пошт def validate\_email(email):

**Функція validate\_email(email)**

Ця функція призначена для валідації формату електронної пошти. Вона приймає один параметр - строку email, перевіряє її відповідність шаблону електронної пошти за допомогою регулярного виразу і повертає булеве значення: True для валідних електронних адрес і False для невалідних.

Спочатку роблю вираз який буде перевіряти структуру пошти. Загалом адрес пошти складається з трьох частин пошта@сервіс.домен Квадратні дужки визначають набір допустимих символів. В першій частині я зробив щоб користувач міг писати будь які символи. В другій та третій частині всі символи можна писати окрім крапки.

^ - Початок перевірки строки.

$ - Кінець перевірки строки.

email\_struckture = r'^[a-zA-Z0-9\_.-][+@[a-zA-Z0-9-]+.[a-zA-Z0-9-]+$](mailto:+@[a-zA-Z0-9-]+.[a-zA-Z0-9-]+$)'

Після чого використовую оператор if, і роблю таку умову if re.match(email\_struckture, email) ця умова перевіряє, чи відповідає строка email заданому виразу email\_struckture. Якщо відповідає, повертається True, інакше False.

if re.match(email\_struckture, email):

return True

else:

return False

**Тест-кейс 1**

Передумова: Параметр email = "example@gmail.com"

Результат: Функція повертає True

**Тест-кейс 2**

Передумова: Параметр email = "invalid-email"

Результат: Функція повертає False

**Тест-кейс 3**

Передумова: Параметр email = [test.email@domain.com](mailto:test.email@domain.com)

Результат: Функція повертає True

**Тест-кейс 4**

Передумова: Параметр email = "user@12"

Результат: Функція повертає False