

Лабораторна робота №2. «Робота з рядками»**Завдання**

Завдання 1. Створіть та виведіть змінну `paragraph`, яка містить текст такого вигляду та форматування:

"Python is a great language!", Said Fred. "I do not
ever remember having this much fun before."

Завдання 2. Створіть текстовий рядок з довільним текстом, який складається з 3 абзаців та виведіть його в такому вигляді, щоб слова, які починаються на 'р' (або 'п', якщо текст написано кирилицею) були з великої літери.

Завдання 3. Зв'яжіть будь-яку змінну з рядком: «Ми обов'язково навчимося програмувати». Виведіть із неї такі зрізи:

- 1) виведіть третій символ цього рядка;
- 2) виведіть передостанній символ цього рядка;
- 3) виведіть перші п'ять символів цього рядка;
- 4) виведіть весь рядок, крім останніх двох символів;
- 5) виведіть всі символи з парними індексами;
- 6) виведіть всі символи з непарними індексами, тобто, починаючи з першого символу рядка;
- 7) виведіть символи з індексами, кратними трьом;
- 8) виведіть всі символи в зворотному порядку;
- 9) виведіть всі символи рядка через один в зворотному порядку, починаючи з останнього;
- 10) виведіть довжину цього рядка;
- 11) виведіть чотири символи з центру рядка;
- 12) видаліть друге слово з рядка;
- 13) замініть друге слово на рядок «ніколи не»;
- 14) додайте в кінець рядка «на Python»;
- 15) виведіть останнє слово першим в рядку.

Завдання 4. Створіть рядок `filename` зі значенням `'hello.py'`. Перевірте, чи завершується ім'я файлу суфіксом `'java'`. Визначте індекс підрядка `'py'`. Перевірте, чи починається рядок з підрядка `'world'`.

Завдання 5. Як використовувати методи *split* і *join* для заміни всіх пропусків у рядку *x* дефісами – наприклад, перетворити "this is a test" в "this-is-a-test"?

Завдання 6. Якщо рядок *x* дорівнює "(name, date), \n", чи поверне рядок "name, date" якась з наступних інструкцій?

- a) `x.rstrip ("", ")`
- b) `x.strip ("", \n")`
- c) `x.strip ("\n) (, ")`

Завдання 7. Необхідно перевірити, чи завершується рядок підрядком "rejected". Який метод рядків для цього використовується? Чи можна домогтися того ж результату іншими способами?

Завдання 8. Як знайти позицію останньої букви *p* в слові Mississippi? А після того, як ця позиція буде знайдена, який код використати для видалення тільки цієї літери?

Завдання 9. Що буде містити змінна *x* при виконанні наступних фрагментів коду?

- a) `x = "{1:{0}}".format(3, 4)`
- b) `x = "{0:$>5}".format(3)`
- c) `x = "{a:{b}}".format(a = 1, b = 5)`
- d) `x = "{a:{b}}:{0:$>5}".format(3, 4, a = 1, b = 5, c = 10)`

Завдання 10. Створіть змінну *item*, яка вказує на рядок "car". Створіть змінну *cost*, яка вказує на число 13499.991

Виведіть рядок, в якій значення *item* вирівнюється по лівому краю поля з 10 символів, а *cost* - по правому краю поля з 10 символів з 2 цифрами в дробовій частині і поділом тисяч запитом. Результат повинен виглядати так:

```
'car    13,499.99'
```

Виконайте завдання двома способами: за допомогою *format()* та *f*-рядку.

Для виконання завдань 9 та 10 скористайтесь теоретичною інформацією, наведеною в Додатку А.

При виконанні завдань для виводу на їх результат використовуйте текстові пояснення з використанням *f*-рядків або методу *format()*

Теоретичні відомості.**Розширене форматування рядків. Підстановки**

Ще один спосіб передачі значень в рядок, що форматується, є використання **підстановок** або **спеціальних плейсхолдерів**, на місце яких вставляються певні значення. Для форматування використовуються такі плейсхолдери:

- s: для вставки рядків
- d: для вставки цілих чисел
- f: для вставки дробових чисел. Для цього типу також можна визначити через точку кількість знаків у дробовій частині.
- %: множить значення на 100 і додає знак відсотка
- e: виводить число в експоненційному записі

Загальний синтаксис плейсхолдера такий:

{ : плейсхолдер }

Залежно від плейсхолдера можна додавати додаткові параметри. Наприклад, для форматування чисел *float* можна використовувати таке:

{ : [кількість_символів] [кома] [.число_знаків_в_дробовій_часті] плейсхолдер }

При виклику методу *format()* в нього в якості аргументів передаються значення, які вставляються на місце плейсхолдерів:

```
welcome = "Hello {:s}"
name = "Tom"
formatted_welcome = welcome.format(name)
print(formatted_welcome) # Hello Tom
```

Як результат метод *format()* повертає новий відформатований рядок.

Форматування цілих чисел:

```
source = "{:d} символів"
number = 5
target = source.format(number)
print(target) # 5 символів
```

Якщо форматується число більше 999, то можна вказати у визначенні плейсхолдера, що необхідно використовувати кому як роздільник розрядів:

```
source = "{:,d} символів"
print (source.format (5000)) # 5,000 символів
```

Для чисел типу *float*, перед кодом плейсхолдера після точки можна вказати, скільки знаків у дробовій частині необхідно вивести:

```
number = 23.8589578
print ( "{:.2f}".format (number)) # 23.86
print ( "{:.3f}".format (number)) # 23.859
```

```
print ( "{:.4f}".format (number)) # 23.8590
print ( "{:,.2f}".format (10001.23554)) # 10,001.24
```

Ще один параметр дозволяє встановити мінімальну ширину значення в символах, що форматується:

```
print ( "{:10.2f}".format (23.8589578)) # 23.86
print ( "{:8d}".format (25)) # 25
```

Для виведення відсотків краще скористатися кодом "%":

```
number = .12345
print ( "{:%}".format (number)) # 12.345000%
print ( "{:.0%}".format (number)) # 12%
print ( "{:.1%}".format (number)) # 12.3%
```

Для виведення числа в експоненційному записі застосовується плейсхолдер "e":

```
number = 12345.6789
print ( "{:e}".format (number)) # 1.234568e + 04
print ( "{:.0E}".format (number)) # 1e + 04
print ( "{:.1e}".format (number)) # 1.2e + 04
```

f-рядки

На даний момент для форматування рядків рекомендується використовувати *f*-рядки, які вперше з'явилися у версії Python 3.6.

Щоб створити *f*-рядок, потрібно зробити таке:

- ввести літеру *f* або *F* перед першою лапкою рядку;
- помістити імена змінної або вирази у фігурні дужки ({}), щоб їх значення потрапили до рядка.

Цей спосіб форматування схожий на використання методу *format()*, але тут у рядку формату не задіяні порожні дужки ({}), та позиційні аргументи ({1}).

```
>>> thing = 'wereduck'
>>> place = 'werepond'
>>> f'The {thing} is in the {place}'
'The wereduck is in the werepond'
```

У фігурних дужках можна розміщувати і вирази:

```
>>> f'The {thing.capitalize()} is in the {place.rjust(20)}'
'The wereduck          is in the .....werepond..... '
```

Для *f*-рядків використовується така сама мова форматування (ширина, заповнювач, вирівнювання), як і в методі *format()*. Вирази розміщуються після двокрапки:

```
>>> f'The {thing:>20} is in the {place:.^20}'
'The wereduck is in the .....werepond.....'
```

Починаючи з версії Python 3.8, *f*-рядки дозволяють виводити не тільки значення змінних, а й їхні імена. Це дуже зручно при налагодженні. Ідея полягає в тому, щоб поставити знак `=` після імені змінної, розміщеного у фігурних дужках:

```
>>> f'{thing =}, {place =}'  
thing = 'wereduck', place = 'werepond'
```

Цей прийом також працює і для виразів. Кожен вираз буде виведено повністю:

```
>>> f'{thing[-4:] =}, {place.title() =}'  
thing[-4:] = 'duck', place.title() = 'Werepond'
```

Нарешті, після знака `=` можна поставити двокрапку, за якою слідуватимуть аргументи, такі як довжини та вирівнювання:

```
>>> f'{thing = :>4.4}'  
thing = 'were'
```