# Лабораторна робота 6 Обробка рядків в Python

**Мета роботи:** Ознайомитись з технологією обробки рядків в Python. Побудувати блок-схему алгоритму, розробити програму з використанням функцій і методів обробки рядків

## Рейтинг лабораторної роботи №6

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Вид діяльності студента | Рейтинговий бал | Deadline |
| 1 | Написання коду | 1 | Жовтень |
| 2 | Захист роботи | 1 |
| 3 | Звіт з роботи | 1 |
| 4 | Якість роботи | 1 |
| Разом за роботу | | 4 |  |

## Теоретичний матеріал

1. Рядок - це послідовність символів, укладених в одинарні або подвійні лапки.
2. Рядок може складатися з декілька підрядків, тоді використовуються по три одинарні або три подвійні лапки
3. Нумерація символів в рядку починається з 0.
4. В Python немає символьного типу, тобто типу даних, об'єктами якого є поодинокі символи. Однак Python дозволяє розглядати рядки як об'єкти, що складаються з підрядків довжиною в один і більше символів.
5. Рядок - це впорядкована послідовність елементів. Отже, з неї можна витягувати окремі символи і зрізи.
6. Створивши рядок, можна отримати будь-яку його частину як новий рядок. Ця частина називатиметься **зрізом**.
7. Номери символів в рядку називаються **індексом**.
8. Якщо вказати від'ємне значення індексу, то номер буде відраховуватися з кінця, починаючи з номера -1.
9. Рядки в Python відносяться до категорії **незмінних послідовностей**, тобто всі функції і методи можуть лише створювати новий рядок.

**Зріз (slice)** - витяг з рядка одного символу або деякого фрагмента підрядка або підпослідовності.

**Взяття одного символу рядка**: S [i] – це зріз, що складається з одного символу, який має номер i.

**Зріз з двома параметрами:** S [a: b] повертає підрядок з **b - a** символів, починаючи з символу з індексом **a** до символу з індексом b, не включаючи його.

**Зріз з трьома параметрами:** Якщо задати зріз з трьома параметрами S [a: b: d], то третій параметр задає крок, тобто будуть взяті символи з індексами a, a + d, a + 2 \* d і т. д

### Функції і методи роботи з рядками

|  |  |
| --- | --- |
| **Функція або метод** | **Призначення** |
| **S = 'str'; S = "str";**  **S = '''str'''; S = """str"""** | Літерали рядків |
| **S = "s\np\ta\nbbb"** | Екрановані послідовності |
| **S = r"C:\temp\new"** | Неформатовані рядки (пригнічують екранування) |
| **S = b"byte"** | Рядок байтів |
| **S1 + S2** | Конкатенація (додавання рядків) |
| **S1 \* 3** | Повторення рядка |
| **S[i]** | Звернення за індексом |
| **S[i:j:step]** | Витяг зрізу |
| **len**(S) | Довжина рядка |
| **S.find**(str, [start],[end]) | Пошук підрядка в рядку. Повертає номер першого входження або -1 |
| **S.rfind**(str, [start],[end]) | Пошук підрядка в рядку. Повертає номер останнього входження або -1 |
| **S.index(str, [start],[end])** | Поиск подстроки в строке. Возвращает номер первого вхождения или вызывает ValueError |
| **S.rindex(str, [start],[end])** | Поиск подстроки в строке. Возвращает номер последнего вхождения или вызывает ValueError |
| **S.replace(шаблон, замена)** | Замена шаблона |
| **S.split(символ)** | Разбиение строки по разделителю |
| **S.isdigit()** | Состоит ли строка из цифр |
| **S.isalpha()** | Состоит ли строка из букв |
| **S.isalnum()** | Состоит ли строка из цифр или букв |
| **S.islower()** | Состоит ли строка из символов в нижнем регистре |
| **S.isupper()** | Состоит ли строка из символов в верхнем регистре |
| **S.isspace()** | Состоит ли строка из неотображаемых символов (пробел, символ перевода страницы ('\f'), "новая строка" ('\n'), "перевод каретки" ('\r'), "горизонтальная табуляция" ('\t') и "вертикальная табуляция" ('\v')) |
| **S.istitle()** | Начинаются ли слова в строке с заглавной буквы |
| **S.upper()** | Преобразование строки к верхнему регистру |
| **S.lower()** | Преобразование строки к нижнему регистру |
| **S.startswith(str)** | Начинается ли строка S с шаблона str |
| **S.endswith(str)** | Заканчивается ли строка S шаблоном str |
| **S.join(список)** | Сборка строки из списка с разделителем S |
| **ord(символ)** | Символ в его код ASCII |
| **chr(число)** | Код ASCII в символ |
| **S.capitalize()** | Переводит первый символ строки в верхний регистр, а все остальные в нижний |
| **S.center(width, [fill])** | Возвращает отцентрованную строку, по краям которой стоит символ fill (пробел по умолчанию) |
| **S.count(str, [start],[end])** | Возвращает количество непересекающихся вхождений подстроки в диапазоне [начало, конец] (0 и длина строки по умолчанию) |
| **S.expandtabs([tabsize])** | Возвращает копию строки, в которой все символы табуляции заменяются одним или несколькими пробелами, в зависимости от текущего столбца. Если TabSize не указан, размер табуляции полагается равным 8 пробелам |
| **S.lstrip([chars])** | Удаление пробельных символов в начале строки |
| **S.rstrip([chars])** | Удаление пробельных символов в конце строки |
| **S.strip([chars])** | Удаление пробельных символов в начале и в конце строки |
| **S.partition(шаблон)** | Возвращает кортеж, содержащий часть перед первым шаблоном, сам шаблон, и часть после шаблона. Если шаблон не найден, возвращается кортеж, содержащий саму строку, а затем две пустых строки |
| **S.rpartition(sep)** | Возвращает кортеж, содержащий часть перед последним шаблоном, сам шаблон, и часть после шаблона. Если шаблон не найден, возвращается кортеж, содержащий две пустых строки, а затем саму строку |
| **S.swapcase()** | Переводит символы нижнего регистра в верхний, а верхнего – в нижний |
| **S.title()** | Первую букву каждого слова переводит в верхний регистр, а все остальные в нижний |
| **S.zfill(width)** | Делает длину строки не меньшей width, по необходимости заполняя первые символы нулями |
| **S.ljust(width, fillchar=" ")** | Делает длину строки не меньшей width, по необходимости заполняя последние символы символом fillchar |
| **S.rjust(width, fillchar=" ")** | Делает длину строки не меньшей width, по необходимости заполняя первые символы символом fillchar |
| **S.format(\*args, \*\*kwargs)** | [Форматирование строки](https://pythonworld.ru/osnovy/formatirovanie-strok-metod-format.html) |

#### Приклад програми1 обробки рядків. HIPO діаграма

#### Умова

Ввести рядок. Витягти ім’я файлу з URL. Видалити пробіли на початку та в кінці рядка. Підрахувати кількість великих та маленьких латинських літер в рядку. Змінити цифрові символи на пробіли. Перевірити складність пароля на наявність цифрових символів, великих літер, довжини не менше 5.

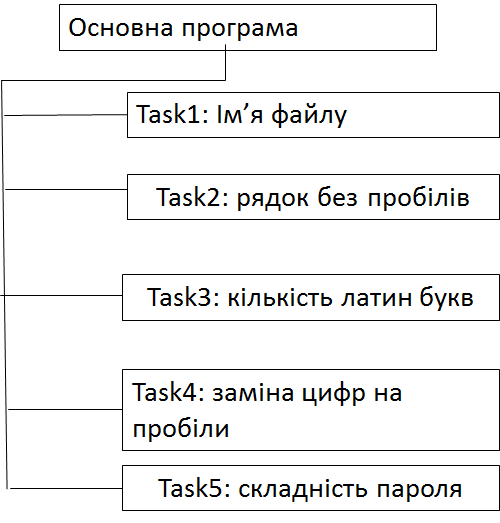


Рис.1. HIPO діаграма виклику функцій

#### Приклад коду

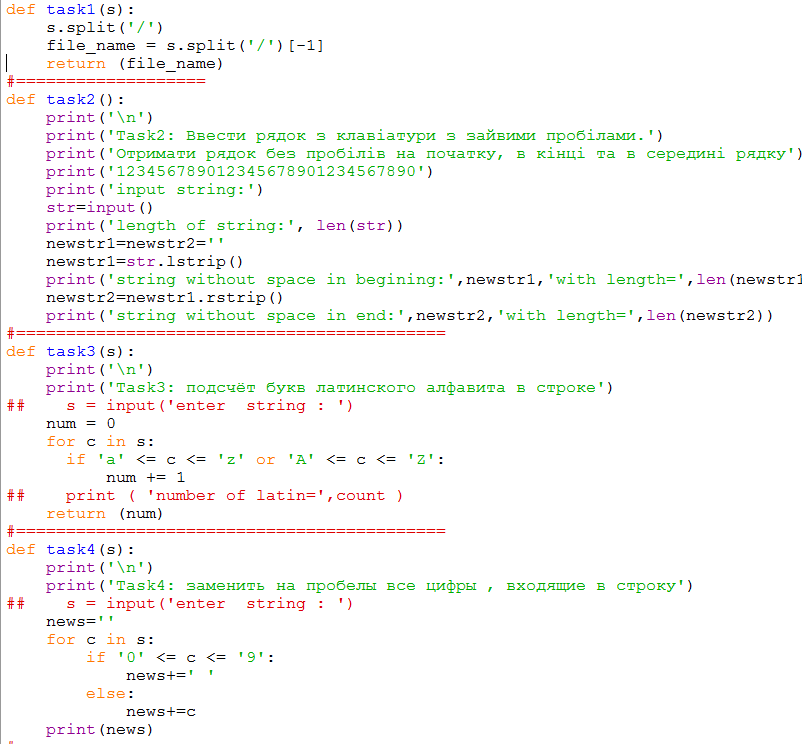


Рис.2. Частина коду програми. Функції.

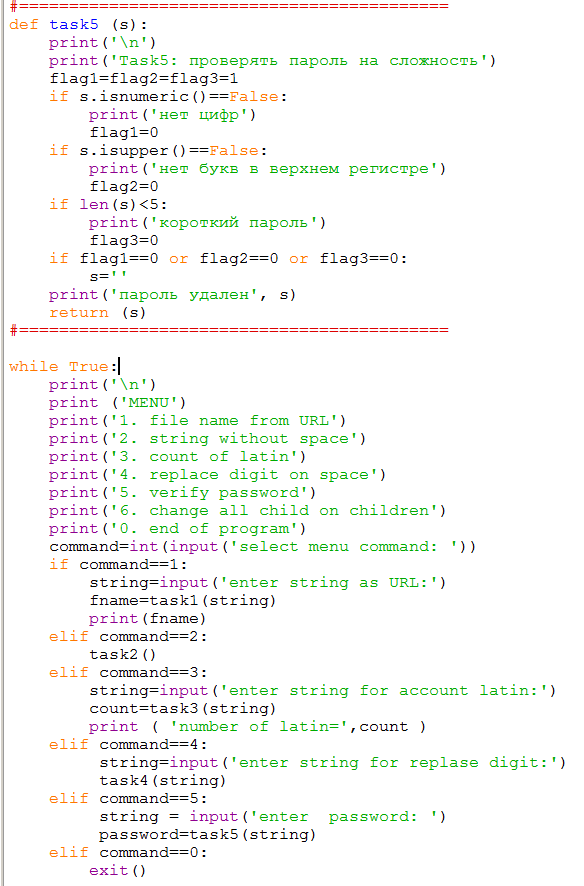


Рис.3. Частина коду програми. Основна програма.

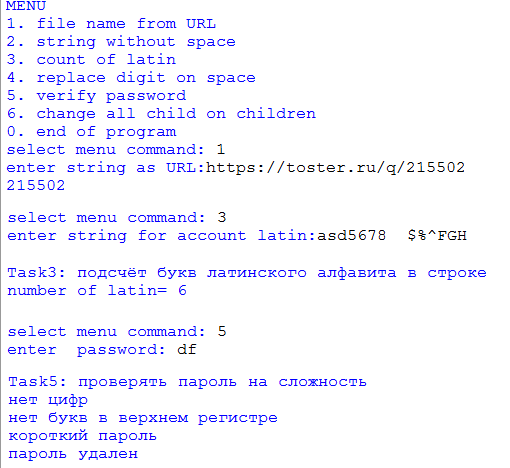


Рис.2. Фрагменти результатів роботи програми.

## Завдання для самостійної роботи (за варіантами)

.

### Варіанти завдань

1. У рядку символів визначити кількість слів і знайти найдовше слово. Вилучити з рядка однолітерні слова та зайві пропуски, коми, крапки. Слова відділяються один від одного довільною кількістю пропусків, ком і крапок.
2. У рядку символів визначити кількість слів, що повторюються, та вилучити дублікати. Слова відокремлюються пробілами.
3. З рядку вилучити всі слова на непарних порядкових позиціях, а слова на парних позиціях надрукувати перевернутими.
4. Ввести два рядки, вилучити з першого рядка всі слова, які зустрічаються у другому рядку.
5. У рядку символів визначити слова-паліндроми, які однаково читаються справа наліво і навпаки. Знайти найбільшу довжину цих слів. Якщо таких слів немає, то з кожного слова на непарній позиції вилучити голосні та надрукуйте їх.
6. У рядку символів визначити слова, що починаються і закінчуються на літери, значення яких уводять з клавіатури. Надрукувати їх у перевернутому вигляді (справа наліво). Визначити частоту кожної голосної літери.
7. У рядку слів визначити кількість слів і надрукувати ті, довжина яких більша за вказану користувачем. Якщо слово починається із заданого користувачем символ, то замінити слово на перше у рядку.
8. Увести рядок символів та отримати новий рядок, слова якого відсортовані за алфавітом. Отримати інший рядок, у якому слова відсортовані за довжиною слів.
9. Розбити на склади згідно з правилами перенесення слів кожне слово на парній позиції у рядку. Слова на непарних позиціях інвертувати (записати у зворотному порядку).
10. Увести рядок, що містить арифметичний вираз (дужки, символи арифметичних операцій, операнди). Визначити коректність виразу (парність дужок, коректність операцій та операндів).
11. Ви перебуваєте в квест-кімнаті по мультфільму WALL-E! Щоб вибратися з кімнати, необхідно виконати такі кроки: а) Дізнайтеся довжину рядка s наведеного тексту і виведіть це значення в консоль. б) Виведіть весь текст в нижньому регістрі.
12. Вводиться рядок, що включає великі та малі літери. Потрібно вивести той самий рядок в одному регістрі, який залежить від того, яких букв більше. При рівній кількості перетворити усі літери в нижній регістр та порахувати кількість голосних.
13. Визначити, скільки разів в тексті зустрічається кожне слово. Слово, яке повторюється найчастіше, розкласти на склади.
14. Ввести рядок і символ. Підрахувати найдовшу послідовність поспіль букв, значення якої введено з клавіатурі. Перетворити рядок, замінивши точками все знаки оклику.
15. Ввести рядок символів, серед яких є дужки, що відкриваються і закриваються. Вивести на екран усі символи, розташовані всередині цих дужок, і підрахувати їх кількість.
16. Ввести рядок, що складається з декількох пропозицій. Кожне Пропозиції закінчується крапкою. Порахувати кількість пропозицій. Перетворити рядок так, щоб кожне речення починалося з великої літери.
17. Роздрукуйте АSCII таблицю, подавши її в 10 колонок. Кожна пара колонок має заголовок «Код» та «Значення символу».
18. Визначте загальну кількість слів у тексті, загальну кількість різних слів (без повторів) та кількість унікальних слів, що зустрічаються тільки один раз. Знайдіть найкоротше слово в тексті.
19. Визначте, з якого слова найчастіше починаються речення у тексті, а також яким найчастіше закінчуються. Визначте частоту появи речень з різною кількістю слів.
20. .Визначте процент води у тексті (це кількість "стоп-слів" поділена на загальну кількість слів). Стоп-слова – це слова, які ігноруються при індексації сторінок пошуковими системами, не несуть смислового навантаження, замінюються маркерами і негативно впливають на якість текстів, знижуючи їх корисність. Списки стоп-слів див. в Інтернеті.
21. Визначте максимальну, мінімальну та середню довжину слів, речень та абзаців у тексті.
22. Ввести речення, що закінчується крапкою. Знайти слово, яке найчастіше повторюється та вивести позиції повторів цього слова.
23. Увести декілька рядків. Об'єднати перші два, встановивши довжину об'єднаного рядка не більше за задану користувачем кількість символів. Символи, що залишилися, приєднати до третього рядка. Повторити операцію для наступної групи рядків.
24. Увести два рядки символів та замінити кожне слово на парній позиції першого рядка на слово, що стоїть на непарній позиції у другому рядку. Кількість слів у рядках може бути різною.
25. Увести рядок символів та отримати новий рядок, слова якого відсортовані за алфавітом. Отримати інший рядок, у якому слова відсортовані за довжиною слів.

## Список літератури

1. <https://github.com/tkovalyuk/Basics-of-programming>
2. **МакГрат М**. Программирование на Python для начинающих. –Москва: Эксмо. – 192 с.
3. **Лутц М.** Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
4. **Лутц М.** Программирование на Python, том I, 2. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. **Васильев А. Н.** Python на примерах. Практический курс ·по программированию. - СПб.:Наука и Техника, 2016. - 432 с.
6. **Доусон М.** Программируем на Python. - СПб.: Питер, 2014. - 416 с.
7. **Рейтц К., Шлюссер Т.** Автостопом по Python. — СПб.: Питер, 2017. — 336 с.