***Робота з рядками у середовищі Python***

Вважають, що програмісти добре знаються в математиці, тому що працюють з числами. Насправді, більшість програмістів працюють з текстовими рядками набагато частіше, ніж з числами.

Будь-яка послідовність символів, укладена в лапки, в Python вважається рядком, при цьому рядки можуть бути укладені як в одинарні, так і в подвійні лапки.

**Рядок (String)** – це складний тип даних, який представляє собою  упорядковані послідовності символів, що використовуються для зберігання і представлення текстової інформації.

**Рядок** можна створити, застосувавши подвійні або одинарні лапки. Однак, початок і кінець рядка повинні обрамлятися однаковим типом лапок.

Наприклад:

**print ("Як тебе не любити, Києве мій!")**

**print ('Як тебе не любити, Києве мій!')**

Щоб ввести більше одного рядка у рядкову величину (багаторядковий рядок), треба використати троє одинарних або подвійних лапок ('''); там, де треба перенести рядок, натиснути ENTER:

**print (''' Як тебе не любити,**

**Києве мій! ''')**

Наприклад: **'125', "мама", 'мама мила раму', 'reed'**

Для рядків існують операції ***конкатенації*** (**+**) і ***дублювання*** (**\***).

*Наприклад:*

|  |
| --- |
|  |
| **a='Іванов'** |
| **b='Дмитро'** |
| **c=a+b # Результат: Іванов Дмитро** |
| **d=a\*3 # Результат:ІвановІвановІванов** |

Якщо всередині рядка потрібно поставити апостроф, то існує два шляхи це зробити:

***І спосіб***

**PC1="комп'ютер"**

***ІІ спосіб***

**PC2='комп\'ютер'**

Застосування потрійних одинарних лапок надає можливість комбінувати подвійні й одинарні лапки в одній стрічці (якщо цього потребує текст) без синтаксичних помилок.

Наприклад, **text='''Він сказав: "Доброго здоров'я! До п'ятниці закінчіть роботу." '''**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **-11** | **-10** | **-9** | **-8** | **-7** | **-6** | **-5** | **-4** | **-3** | **-2** | **-1** |
| ***І*** | ***н*** | ***ф*** | ***о*** | ***р*** | ***м*** | ***а*** | ***т*** | ***и*** | ***к*** | ***а*** |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |

Як бачимо, усередині тексту ми без проблем використовуємо одинарні і подвійні лапки. Сам же текст починається і завершується потрійними лапками **'''**.

***Індексація символів у рядку***

Всі елементи рядка мають свій порядковий номер починаючи з нуля – індекс. Наприклад:

У послідовностях важливий порядок символів, у кожного символу в рядку є унікальний порядковий номер – індекс.

Індексація починається не з одиниці, а з нуля. Дозволено витягувати символи, починаючи відлік з кінця. У цьому випадку відлік починається з -1 (останній символ).

Можна звертатися до конкретного символу в рядку і отримувати його з допомогою оператора індексування, який представляє собою квадратні дужки з номером символу в них.

|  |  |
| --- | --- |
| **tday = 'Ранок, день, ніч'** |  |
| **print(tday[1])** | **#Результат: 'o'** |
| **print( tday[4])** | **#Результат: 'i'** |

Індексація починається не з одиниці, а з нуля. Тому вираз **print(tday [1])** призвів до витягнення другого символу.

Коли потрібно витягти перший символ, то оператор індексування повинен виглядати так:  **tday [0]**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **-16** | **-15** | **-14** | **-13** | **-12** | **-11** | **-10** | **-9** | **-8** | **-7** | **-6** | **-5** | **-4** | **-3** | **-2** | **-1** |
| ***р*** | ***а*** | ***н*** | ***о*** | ***к*** | ***,*** |  | ***д*** | ***е*** | ***н*** | ***ь*** | ***,*** |  | ***н*** | ***і*** | ***ч*** |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |

Для того, щоб отримати один символ рядка, задається індекс цього символа у рядку всередині квадратних дужок після імені рядка.

Для читання рядка з клавіатури використовують вже відому нам команду **S=input()**

|  |  |
| --- | --- |
| **tday = 'ранок, день, ніч'** |  |
| **print( tday[0])** | **#Результат: 'р'** |
| **print(tday[-1])** | **#Результат: 'ч'** |
| **print(tday[-3])** | **#Результат: 'н'** |

Необхідно пам'ятати, що рядки в Python відносяться до категорії незмінних послідовностей, тобто всі функції і методи можуть лише створювати новий рядок.

Якщо записати **S[0]=** **'5',** то програма видасть помилку:

**TypeError: 'str' object does not support item assignment**

***Зрізи у рядках***

За допомогою рядкових зрізів з рядка можна «витягти» не лише один символ, але й підрядок (частину рядка). Для цього у квадратних дужках зрізу вказують початок зрізу, кінець, а також крок зрізу, якщо такий є: **S[x:y:step]**, де **x** – це початок зрізу, **y** – кінець зрізу (сам символ в зріз не входить!), **step** – крок зрізу.

*Наприклад:*

|  |  |
| --- | --- |
| **s = 'Спартак'** |  |
| **s[0:6]** | **#Результат: 'Спарта'** |
| **s[1:6]** | **#Результат: 'парта'** |
| **s[::-1]** | **#Результат: 'катрапС'** |
| **s[2::2]** | **#Результат: 'аа'** |

Якщо відсутній перший індекс, то зріз береться від початку до другого індексу; за відсутності другого індексу, зріз береться від першого індексу до кінця рядка.

Запис **s[::-1]** означає, що здійснюється **реверс** рядка (запис у зворотному напрямі з кроком 1).

***Довжина рядка***

Існує спеціальна функція **len()**, яка дозволяє виміряти довжину рядка. Результатом виконання даної функції є число, що показує кількість символів в рядку: **print(len('Hello, World!'))**

Результатом виконання цієї команди буде число 13 (кількість символів у рядку).

***Операції з рядками***

Функція **int** переводить рядок у послідовність цілих чисел.

Функція **float** переводить рядок у послідовність дійсних чисел.

**str(s)** – переклад будь-якого об'єкта в рядковий тип даних.

**len(s)** – визначення довжини тексту.

**min(s)** – отримання мінімального значення з кодової таблиці ASCII.

**max(s)** – отримання максимального значення з кодової таблиці ASCII.

**St1 in St2** – входження підрядка **St1** в **St2**. Поверне True, якщо St1 присутній в початковому тексті.

**St1 not in St2** – перевіряє відсутність **St1** в **St2** і повертає True при вірності твердження.

**S.find(str,[start],[end])** – шукає номер першого входження рядка **str** в рядок **S**, якщо присутні параметри **start** і **end**, то шукає лише на вказаному проміжку. Якщо рядка **str** в рядку **S** немає, то результат виконання -1.

**S.rfind(str,[start],[end])** – шукає номер останнього входження рядка **str** в рядок **S**, якщо присутні параметри **start** і **end**, то шукає лише на вказаному проміжку. Якщо рядка **str** в рядку **S** немає, то результат виконання -1.

**S.replace(a, b)** – заміна шаблону тексту **a** на **b** в рядку **S**.

**S.split(t)** – розбиває рядок **S** розділювачем **t**.

**S.upper()** – переводить всі символи рядка у верхній регістр.

**S.lower()** – переводить всі символи рядка у нижній регістр.

**S.capitalize()** – переводить перший символ рядка у верхній регістр, а всі останні – у нижній.

**S.swapcase()** – переводить всі символи з верхнього регістру у нижній і навпаки.

**S.title()** – записує кожне слово рядка з великої літери.

**sorted(s)** – впорядковує елементи рядка в порядку зростання.

**S.lstrip([chars])** — видалення символів пробілів на початку рядка;

**S.rstrip([chars])** — видалення символів пробілів в кінці рядка;

**S.strip([chars])** — видалення символів пробілів на початку і в кінці рядка.

|  |  |
| --- | --- |
| *Вхідні дані* | *Вихідні дані* |
| **Hello world!** | **2** |

**Приклад завдання:** дано рядок символів. Знайти кількість слів у даному рядку. Слова розділяються довільною кількістю пропусків.

**Практичне завдання.**

1. Створіть і збережіть пустий файл для подальшого виконання.
2. Присвойте змінній **X** будь-який рядок, що складається з не менше ніж з 8 символів (наприклад, «програмування»).
3. Продублюйте рядок 10 разів. Результат присвойте іншій змінній (наприклад, **Y**).
4. Вийміть з рядка:

* перший символ;
* останній символ;
* третій символ з початку і третій з кінця.

1. Виміряйте довжину вашого рядка.
2. Присвойте змінній довільний рядок довжиною 10-15 символів
3. Отримайте з неї наступні зрізи:

* перші вісім символів;
* чотири символи з центру рядка;
* символи з індексами, кратними трьом.

1. Використовуючи метод зрізів, створіть програмний код, який зі слова «Інформатика» створить слова:

* «форма»;
* «формат».

1. Створіть програмний код, який буде:

* визначати довжину рядка з вашим іменем;
* знаходити мінімальний символ в вашому імені;
* знаходити максимальний символ в вашому імені.

1. (*високий рівень*): створіть програмний код, який перевірятиме на паліндромність (паліндром – речення, що читається однаково в прямому та зворотному напрямі). Використайте метод зрізів **[::-1]** та метод **replace** для видалення усіх пробілів з рядка. Для прикладу використайте паліндром: *А роза упала на лапу Азора*.

***Контрольні запитання:***

а) Що таке рядок та рядковий тип даних? б) Як здійснюється індексація символів у рядку; в) Як можна записати апостроф чи лапки усередині рядкової величини? г) Як вилучити потрібний символ з рядка? д) Що таке зрізи і як їх виконують? е) Як дізнатися довжину рядка? є) Які операції і за допомогою яких методів (функцій) можна виконувати над рядками?