Лабораторная работа № 8

«Строки и обработка текстовой информации»

**Цель работы:** *приобрести навыки работы с Python-строками и закрепить их на примере разработки интерактивных приложений.*

**Основное задание**

Написать программу «Анаграммы» («*Anagrams»*). Суть игры заключается в следующем: формируется группа слов в виде кортежа (*tuple*), программа случайным образом выбирает одно из слов, и случайным образом переставляет в нём буквы, а затем предоставляет пользователю (игроку). Цель игрока – угадать выбранное и «перемешанное» программой слово.

**Индивидуальное задание**

В соответствии с заданием своего варианта выполнить задания из Приложения «*C*» «*Строки*».

**Дополнительное задание**

Модифицировать программу «Анаграммы» («*Anagrams»*) таким образом, чтобы в ней были уровни прохождения игры. К примеру, в игре «Анаграммы» можно реализовать 10 уровней, где на первых уровнях предлагаются простые слова. При этом от уровня к уровню слова становятся длиннее, и растёт их уровень сложности. Дополнительно можно добавить к игре ограниченное количество подсказок и попыток угадывания.

**Требования к выполнению**

1. Программа должна обязательно быть снабжена комментариями, в которых необходимо указать краткое предназначение программы, номер лабораторной работы и название, версию программы, ФИО разработчика и дату разработки;
2. В программах, предусмотреть возможность их повторного выполнения, а также защиты от ввода некорректных пользовательских данных. Для этого рекомендуется разработать отдельные функции;
3. Разработанные основные функции бизнес-логики необходимо разместить в отдельном модуле, а затем подключить в другом модуле, где будет происходить вызов данных функций;
4. При разработке программ рекомендуется придерживаться принципа единственной ответственности;
5. Каждая программа должна быть снабжена дружелюбным и интуитивно понятным интерфейсом.

**Контрольные вопросы**

1. Чем характеризуется строковый тип данных в *Python*?
2. Какие есть способы объявления строк в *Python*?
3. Что такое неизменяемость строк?
4. Зачем нужна индексация строк, и как её использовать?
5. Зачем и как используются срезы строк?
6. Какие основные операторы используются для работы со строками?
7. Какие основные встроенные функции класса str используются для работы со строками?
8. Как объявить шаблонную строку, и какие спецификаторы типов (операторы форматирования) в этой строке можно использовать?
9. Что необходимо прописать в исходном коде программы, если нужно использовать различные кодировки для отображения текстовой информации?
10. Как создать всплывающую подсказку с описанием вызываемой функции?

Строки в *Python* и обработка текстовой информации

*Строка* – это *неизменяемая* (*immutable*) *последовательность* символов. Доступ к одному символу можно получить с помощью оператора квадратных скобок (рисунок 8.1).

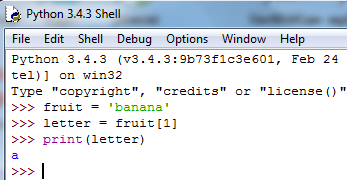


Рис. 8.1. Доступ к одному символу

Выражение в скобках называется *индексом* (*index*). Индекс – это смещение от начала строки; смещение для первого символа – нуль. Таким образом, нумерация в строках начинается с нуля. В качестве индекса можно использовать любое целочисленное выражение, включая переменные или операторы. Можно использовать отрицательные индексы, которые считаются с конца. fruit[-1] выдаст последний символ, fruit[-2] – второй с конца строки и так далее.

На рисунке 8.2 приводится встроенная функцияlen(), которая возвращает число символов в строке.

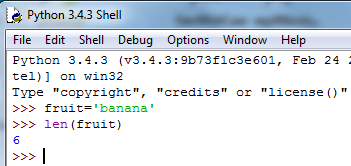


Рис. 8.2. Встроенная функцияlen()

ОБХОД СТРОКИ С ПОМОЩЬЮ ЦИКЛА

Множество алгоритмов включают посимвольную обработку строк, которая, как правило, начинается с начала строки. Такая обработка называется *обходом* (*traversal*). Обход может осуществляться с помощью различных циклов.

На рисунке 8.3 представлен обход с помощью цикла while.

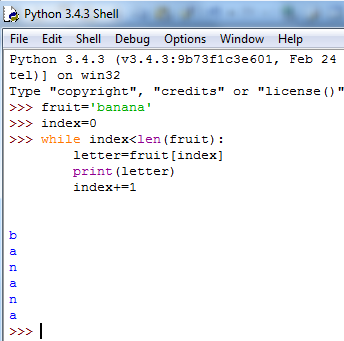


Рис. 8.3. Обход с помощью цикла while

Когда позиция символа не важна, а важно только его значение, то предпочтительнее использовать обход с помощью цикла for (рисунок 8.4).

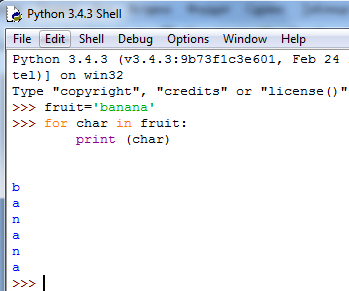


Рис. 8.4. Обход с помощью цикла for

СРЕЗ СТРОКИ

*Срезом* (*slice*) называется часть строки. Оператор [*n*:*m*] возвращает часть строки от *n*-ого символа до *m*-ого символа, включая первый, но исключая последний. Если опустить первый индекс, то срез будет начинаться с начала строки. Если опустить второй индекс – срез завершится в конце строки (рисунок 8.5).

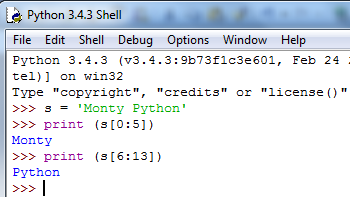


Рис. 8.5. Срез строки

Если первый индекс не меньше второго, то результатом будет *пустая строка*, если первый и второй индексы отсутствуют – выводится вся строка целиком.

Логический операторin принимает две строки и возвращает True, если первая является подстрокой второй (рисунок 8.7).

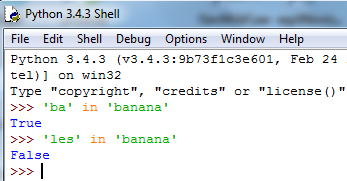


Рис. 8.6. Логический операторin

В *Python* можно сравнивать строки (рисунок 8.7). Все буквы верхнего регистра имеют приоритет перед буквами нижнего, буквы – перед цифрами. При проверке равенства строк можно преобразовать строку в стандартный формат, например, нижний регистр.

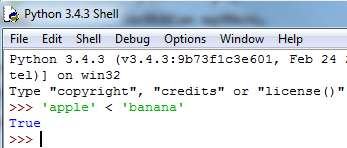


Рис. 8.7. Сравнение строк

СТРОКОВЫЕ МЕТОДЫ

Строка является объектом в *Python*. Объекты содержат данные (фактически саму строку) и методы, которые являются функциями, встроенными в объект и доступными для любого *экземпляра* (*instance*) объекта. Функция dir()выводит список доступных методов для объекта, вводимого в качестве параметра (рисунок 8.8).

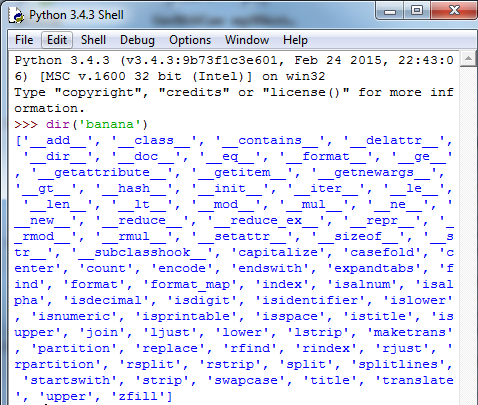


Рис. 8.8. Функция dir()

Метод вызывается аналогично функции, принимает аргументы и возвращает значение. Однако метод и функция различаются синтаксисом: метод вызывается путем добавления имени метода к имени переменной с использованием точки в качестве разделителя. Синтаксис отдельных методов приводится в таблице 8.1.

Таблица 8.1 – Строковые методы

|  |  |  |
| --- | --- | --- |
| *№* | *Метод* | *Описание* |
| 1. | str.upper() | Преобразование строки str к верхнему регистру. |
| 2. | str.lower() | Преобразование строки str к нижнему регистру. |
| 3. | str.find(str2,n) | Возвращает индекс, с которого начинается вхождение строки str2 в строку str. Необязательный параметр n – индекс, с которого начинается поиск вхождения. |
| 4. | str.strip() | Устранение лишних пробелов. |
| 5. | str.capitalize() | Преобразовывает из строчной в прописную первую букву строки *str.* |
| 7. | str.count(str,beg=0,  end=len(string)) | Подсчитывает количество вхождений подстроки str в оригинальной строке (beg – стартовая позиция в строке поиска, end – конечная позиция) |
| 9. | str.endswith(suffix,  beg=0,end=len(string)) | Возвращает булевское значение True, если оригинальная строка заканчивается последовательностью suffix, иначе возвращается False*.* |
| 10. | str.startswith(suffix, beg=0,end=len(string)) | Возвращает булевское значение True, если оригинальная строка начинается с последовательности suffix, иначе возвращается False. |

**ПРИЛОЖЕНИЕ *C* «Строки»**

**Задание 1.** *В соответствии с заданием своего варианта составить программу для анализа текста, вводимого с клавиатуры.*

**Задание 2.** *Дана строка текста, в которой слова разделены пробелами и запятыми. В соответствии с заданием своего варианта составить программу для анализа строки, инициализированной в коде программы:* «*So she was considering in her own mind, as well as she could, for the hot day made her feel very sleepy and stupid, whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.*» *Если не оговорено иное, то регистр букв при решении задачи не имеет значения.*

Таблица *C*1. Индивидуальные задания.

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Задания** | **Условие** |
| **1** | **1** | В строке, вводимой с клавиатуры, подсчитать количество заглавных английских букв. |
| **2** | а) определить количество слов в строке;  б) найти самое длинное слово и его порядковый номер;  в) вывести каждое четное слово. |
| **2** | **1** | Определить, является ли введенная с клавиатуры строка двоичным числом. |
| **2** | а) определить количество строчных букв;  б) найти первое слово, содержащее букву '*v*' и его номер;  в) вывести строку, исключив из нее слова, начинающиеся с '*s*'. |
| **3** | **1** | Определить, является ли введенная с клавиатуры строка шестнадцатеричным числом. |
| **2** | а) определить количество слов в строке и вывести на экран все слова, количество букв у которых четное;  б) найти самое короткое слово, которое начинается на '*a*';  в) вывести повторяющиеся слова. |
| **4** | **1** | В строке, вводимой с клавиатуры, подсчитать количество пробельных символов. |
| **2** | а) определить количество слов, длина которых меньше 5;  б) найти самое короткое слово, заканчивающееся на букву '*d*';  в) вывести все слова в порядке убывания их длин. |
| **5** | **1** | В строке, вводимой с клавиатуры, подсчитать количество слов, начинающихся со строчной буквы. |
| **2** | а) определить, сколько слов имеют минимальную длину;  б) вывести все слова, за которыми следует запятая;  в) найти самое длинное слово, которое заканчивается на '*y*'. |
| **6** | **1** | В строке, вводимой с клавиатуры, подсчитать количество символов, отличных от пробельных. |
| **2** | а) определить число слов, заканчивающихся на гласную букву;  б) найти среднюю длину слов в строке, округлив результат до целого числа, и вывести все слова, которые имеют такую длину, или сообщение «Слов длиной n символов в строке нет»;  в) вывести каждое пятое слово. |
| **7** | **1** | В строке, вводимой с клавиатуры, подсчитать количество букв нижнего регистра и цифр. |
| **2** | а) определить число слов, которые начинаются с согласной;  б) найти слова, содержащие две одинаковые буквы подряд и их порядковые номера;  в) вывести слова в алфавитном порядке. |
| **8** | **1** | В строке, вводимой с клавиатуры, подсчитать количество цифр. |
| **2** | а) определить число слов, длина которых равна 3 символа;  б) найти слова, у которых количество гласных равно количеству согласных и их порядковые номера;  в) вывести слова в порядке убывания их длин. |
| **9** | **1** | В строке, вводимой с клавиатуры, подсчитать количество пробелов и знаков пунктуации. |
| **2** | а) определить число слов, начинающихся или заканчивающихся на гласную букву;  б) определить, сколько раз повторяется каждый символ;  в) вывести в алфавитном порядке слова, идущие после запятой. |
| **10** | **1** | В строке, вводимой с клавиатуры, подсчитать количество символов, лежащих в диапазоне от '*g*' до '*o*'. |
| **2** | а) определить число слов, ограниченных пробелами;  б) определить, сколько раз повторяется каждая буква;  в) вывести по алфавиту словосочетания, отделенные запятыми. |
| **11** | **1** | В строке, вводимой с клавиатуры, подсчитать количество заглавных гласных английских букв. |
| **2** | а) определить количество слов в строке;  б) найти самое длинное слово и его порядковый номер;  в) вывести каждое нечетное слово. |
| **12** | **1** | Определить, является ли введенная с клавиатуры строка двоичным числом. |
| **2** | а) определить количество заглавных строчных букв;  б) найти первое слово, содержащее букву '**z**' и его номер;  в) вывести строку, исключив из нее слова, начинающиеся с '*a*'. |
| **13** | **1** | Определить, является ли введенная с клавиатуры строка восьмеричным числом. |
| **2** | а) определить количество слов в строке и вывести на экран все слова, количество букв у которых нечетное;  б) найти самое короткое слово, которое начинается на букву '*i*';  в) вывести повторяющиеся слова. |
| **14** | **1** | В строке, вводимой с клавиатуры, подсчитать количество пробельных символов и запятых. |
| **2** | а) определить число слов, длина которых меньше 7 символов;  б) найти самое короткое слово, заканчивающееся на букву '*a*';  в) вывести все слова в порядке убывания их длин. |
| **15** | **1** | В строке, вводимой с клавиатуры, подсчитать количество слов, начинающихся со строчной согласной буквы. |
| **2** | а) определить, сколько слов имеют максимальную длину;  б) вывести все слова, за которыми следует запятая или точка;  в) найти самое длинное слово, которое заканчивается на '*е*'. |
| **16** | **1** | В строке, вводимой с клавиатуры, подсчитать количество знаков пунктуации. |
| **2** | а) определить число слов, заканчивающихся на согласную;  б) найти среднюю длину слов в строке, округлив результат до целого числа, и вывести все слова, которые имеют такую длину, или сообщение «Слов длиной *n* символов в строке нет»;  в) вывести каждое седьмое слово. |
| **17** | **1** | В строке, вводимой с клавиатуры, подсчитать количество букв латинского алфавита и цифр. |
| **2** | а) определить число слов в строке, начинающихся с гласной;  б) найти слова, содержащие две одинаковые буквы подряд и их порядковые номера;  в) вывести слова в алфавитном порядке. |
| **18** | **1** | В строке, вводимой с клавиатуры, подсчитать количество цифр и гласных букв. |
| **2** | а) определить число слов, длина которых равна 4 символа;  б) найти слова, у которых количество гласных равно количеству согласных и их порядковые номера;  в) вывести слова в порядке убывания их длин. |
| **19** | **1** | В строке, вводимой с клавиатуры, подсчитать количество пробелов, цифр и знаков пунктуации. |
| **2** | а) определить число слов, начинающихся или заканчивающихся на гласную букву;  б) определить, сколько раз повторяется каждый символ;  в) вывести в алфавитном порядке слова, идущие после запятой. |
| **20** | **1** | В строке, вводимой с клавиатуры, подсчитать количество символов, лежащих в диапазоне от '*f*' до '*y*'. |
| **2** | а) определить количество слов, заключенных в кавычки;  б) определить, сколько раз повторяется каждая буква;  в) вывести в алфавитном порядке все словосочетания, отделенные запятыми. |
| **21** | **1** | В строке, вводимой с клавиатуры, подсчитать количество заглавных английских букв и цифр. |
| **2** | а) определить количество слов в строке;  б) найти самое длинное слово и его порядковый номер;  в) вывести каждое нечетное слово. |
| **22** | **1** | Определить, является ли введенная с клавиатуры строка двоичным числом. |
| **2** | а) определить количество строчных букв;  б) найти последнее слово, содержащее букву '*i*' и его номер;  в) вывести строку, исключив из нее слова, начинающиеся с '*i*'. |
| **23** | **1** | Определить, является ли введенная с клавиатуры строка шестнадцатеричным числом. |
| **2** | а) определить количество слов, состоящих из прописных букв ;  б) найти самое длинное слово, которое начинается на букву '*l*';  в) вывести повторяющиеся слова. |
| **24** | **1** | В строке, вводимой с клавиатуры, подсчитать количество пробельных символов и апострофов. |
| **2** | а) определить число слов, длина которых меньше 6 символов;  б) найти самое короткое слово, заканчивающееся на букву '*w*';  в) вывести все слова в порядке возрастания их длин. |
| **25** | **1** | В строке, вводимой с клавиатуры, подсчитать количество слов, начинающихся со строчной согласной буквы. |
| **2** | а) определить, сколько слов имеют минимальную длину;  б) вывести все слова, за которыми следует точка;  в) найти самое длинное слово, которое заканчивается на '*r*'. |