ТЕМА 4 · СТРОКИ · PYTHON 3

Практика: методы строк

D1–D16 · от одного метода к обработке текстовых данных

Перед началом

Проверить решения D1–D16 →

Сохраняй решения в файлы `d1.py`, `d2.py` и так далее. Во всех задачах вводится **одна строка** через `input()` без приглашения. Длина входной строки — от 0 до 1000 символов; дополнительные ограничения указаны в задачах. Пустой ввод — это нажатие Enter. Завершающий перевод строки не является частью входных данных.

Печатай только то, что требует условие. Регистр в ответах `YES`, `NO`, `EMPTY`, `START`, `STOP`, `UNKNOWN` важен. Не добавляй `strip()` автоматически: иногда пробелы запрещены и должны привести к отказу, а иногда их нужно сохранить.

> Это учебная практика. Выполняй задачи в IDLE и проверяй .py-файлы в автотестере. Можно отправить часть решений и продолжить позже. Сдавай файлы способом, который укажет учитель. Подсказки раскрываются отдельно; готовых решений в ученическом комплекте нет.

Маршрут	Задачи
После урока 1	D1–D3
После урока 2	D4–D8
После урока 3	D9–D12
Дополнительная практика	D13–D16

> В примерах показана **запись текста в кавычках**, чтобы были видны пробелы. Кавычки не вводи и не печатай. `''` означает пустую строку; `\n` внутри записи вывода означает переход на следующую строку; `\t` — табуляцию. Например, `'a  b'` содержит два пробела.

D1. Имя команды

Файл: `d1.py` · База · урок 1

Вводится одна строка — название команды. Удали пробельные символы только по краям и переведи буквы в нижний регистр. Пробелы внутри сохрани.

Выходные данные

Обработанное название. Если оно пустое, выведи пустую строку.

Пример 1

Ввод · запись строки

```
'  PyThOn  '
```

Вывод · запись текста

```
'python'
```

Пример 2

Ввод · запись строки

```
'  Красный  Робот '
```

Вывод · запись текста

```
'красный  робот'
```

Подсказка / самопроверка

Нужно два преобразования. У каждого результата тип str; можно вызвать следующий метод через точку.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.

D2. Расшифруй сообщение

Файл: `d2.py` · База · урок 1

Вводится строка сообщения. Все символы подчёркивания _ обозначают пробелы. Замени каждое подчёркивание одним обычным пробелом; остальные символы сохрани.

Выходные данные

Расшифрованное сообщение; число пробелов после замены не сокращай.

Пример 1

Ввод · запись строки

```
'Робот_готов'
```

Вывод · запись текста

```
'Робот готов'
```

Пример 2

Ввод · запись строки

```
'A__B'
```

Вывод · запись текста

```
'A  B'
```

Подсказка / самопроверка

В replace первый аргумент — что ищем, второй — на что меняем. Ограничение числа замен не нужно.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.

D3. Табличка на дверь

Файл: `d3.py` · База · урок 1

Вводится строка для таблички. Убери пробельные края, переведи буквы в верхний регистр и замени только первый обычный пробел внутри результата на дефис. Остальные внутренние пробелы сохрани.

Выходные данные

Текст таблички после трёх преобразований.

Пример 1

Ввод · запись строки

```
'  кабинет робототехники '
```

Вывод · запись текста

```
'КАБИНЕТ-РОБОТОТЕХНИКИ'
```

Пример 2

Ввод · запись строки

```
'клуб юных роботов'
```

Вывод · запись текста

```
'КЛУБ-ЮНЫХ РОБОТОВ'
```

Подсказка / самопроверка

Сначала убери края, иначе первой заменой окажется случайный пробел перед текстом. Третий аргумент replace равен 1.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.

D4. Упоминания робота

Файл: `d4.py` · База · урок 2

Вводится строка. Посчитай неперекрывающиеся вхождения подстроки «робот» без учёта регистра. Вхождение внутри длинного слова тоже учитывается; искать отдельные слова не требуется.

Выходные данные

Одно целое число — число вхождений.

Пример 1

Ввод · запись строки

```
'Робот и РОБОТ'
```

Вывод · запись текста

```
'2'
```

Пример 2

Ввод · запись строки

```
'робототехника'
```

Вывод · запись текста

```
'1'
```

Подсказка / самопроверка

Сначала приведи текст к одному регистру. count возвращает число, его и печатаем.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.

D5. Где собачка?

Файл: `d5.py` · База · урок 2

Вводится строка. Найди индекс первого символа @, считая позиции с нуля. Если такого символа нет, выведи −1. Это задача о поиске символа, а не о проверке настоящего адреса почты.

Выходные данные

Индекс первого @ или −1.

Пример 1

Ввод · запись строки

```
'user@mail.ru'
```

Вывод · запись текста

```
'4'
```

Пример 2

Ввод · запись строки

```
'@robot'
```

Вывод · запись текста

```
'0'
```

Подсказка / самопроверка

Нужен метод, который вернёт −1 при отсутствии и не вызовет ошибку.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.

D6. Python-файл

Файл: `d6.py` · База · урок 2

Вводится имя файла. Удали пробельные края, затем проверь, заканчивается ли имя на .py без учёта регистра. В этой учебной задаче других требований к имени нет.

Выходные данные

YES, если условие выполнено, иначе NO.

Пример 1

Ввод · запись строки

```
'  Lesson.PY '
```

Вывод · запись текста

```
'YES'
```

Пример 2

Ввод · запись строки

```
'task.py.txt'
```

Вывод · запись текста

```
'NO'
```

Подсказка / самопроверка

Подстрока .py должна стоять в конце после очистки и смены регистра.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.

D7. Код пропуска

Файл: `d7.py` · База · урок 2

Вводится строка из латинских/русских букв, обычных цифр 0–9, пробелов и знаков ASCII. Других символов-цифр Unicode во входе нет. Корректный код содержит ровно четыре цифры 0–9. Пробелы не удаляй. Ведущие нули разрешены.

Выходные данные

YES для корректного кода, иначе NO.

Пример 1

Ввод · запись строки

```
'0072'
```

Вывод · запись текста

```
'YES'
```

Пример 2

Ввод · запись строки

```
' 072'
```

Вывод · запись текста

```
'NO'
```

Подсказка / самопроверка

Нужны одновременно два условия: длина и состав. Код остаётся строкой.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.

D8. Учебный логин

Файл: `d8.py` · Закрепление · урок 2

Вводится строка. Сначала убери пробельные края. Логин допустим, если его длина от 3 до 12 включительно, в нём только буквы, цифры и подчёркивания _, а также есть хотя бы одна буква или цифра. Буквы/цифры понимаются так же, как в isalnum(). Внутренние пробелы запрещены.

Выходные данные

YES для допустимого логина, иначе NO.

Пример 1

Ввод · запись строки

```
'  bot_7 '
```

Вывод · запись текста

```
'YES'
```

Пример 2

Ввод · запись строки

```
'___'
```

Вывод · запись текста

```
'NO'
```

Подсказка / самопроверка

Проверять длину нужно у очищенного логина, до удаления подчёркиваний. Для проверки состава временно убери только _.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.

D9. Сколько слов?

Файл: `d9.py` · База · урок 3

Вводится строка. Словом считаем каждый элемент разбиения по пробельным промежуткам: знаки препинания остаются частью слова. Посчитай слова; строка может быть пустой или состоять из пробелов.

Выходные данные

Число слов.

Пример 1

Ввод · запись строки

```
'  робот   едет домой  '
```

Вывод · запись текста

```
'3'
```

Пример 2

Ввод · запись строки

```
'   '
```

Вывод · запись текста

```
'0'
```

Подсказка / самопроверка

Нужен split без явного разделителя: он не создаёт пустые слова из лишних пробелов.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.

D10. Адрес страницы кружка

Файл: `d10.py` · База · урок 3

Вводится название кружка. Переведи буквы в нижний регистр, выдели слова по пробельным промежуткам и соедини их одним дефисом. Знаки препинания не меняй. Для пустого названия результат пустой.

Выходные данные

Строка со словами через дефис без дефисов по краям.

Пример 1

Ввод · запись строки

```
'  Клуб   Роботов  '
```

Вывод · запись текста

```
'клуб-роботов'
```

Пример 2

Ввод · запись строки

```
'Python'
```

Вывод · запись текста

```
'python'
```

Подсказка / самопроверка

Сначала получится список строк, затем снова строка. Разделитель вызывает join.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.

D11. Заполненные поля

Файл: `d11.py` · Закрепление · урок 3

Вводится строка полей, разделённых точками с запятой ;. Поле считается заполненным, если после удаления пробельных краёв в нём остаётся хотя бы один символ. Посчитай заполненные поля. Сами поля могут содержать внутренние пробелы.

Выходные данные

Количество заполненных полей.

Пример 1

Ввод · запись строки

```
'робот;; датчик ; ;мотор;'
```

Вывод · запись текста

```
'3'
```

Пример 2

Ввод · запись строки

```
' ; ; '
```

Вывод · запись текста

```
'0'
```

Подсказка / самопроверка

Сначала разбей по ;. Каждый полученный элемент очисти отдельно и реши, нужно ли увеличить счётчик.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.

D12. Подпись участника

Файл: `d12.py` · Закрепление · урок 3

Вводится имя и фамилия — ровно два непустых слова из букв в правильном регистре, например «Анна Петрова». Между словами и по краям может быть несколько пробелов. Выведи фамилию, пробел, первую букву имени в верхнем регистре и точку. Гарантируется наличие двух слов.

Выходные данные

Подпись в формате Фамилия И.

Пример 1

Ввод · запись строки

```
'Анна Петрова'
```

Вывод · запись текста

```
'Петрова А.'
```

Пример 2

Ввод · запись строки

```
'  иван   Иванов  '
```

Вывод · запись текста

```
'Иванов И.'
```

Подсказка / самопроверка

parts[0] — первое слово; parts[0][0] — первая буква этого слова. Разделение гарантирует два элемента по условию.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.

D13. Расширение после последней точки

Файл: `d13.py` · Дополнительно

Вводится только имя файла, без пути. Удали пробельные края и найди последнюю точку. Расширением считаем непустой фрагмент после неё, если перед ней есть хотя бы один символ. Если последняя точка стоит первой или последней либо точек нет, расширения нет. Это учебное правило, не полный разбор файловых систем.

Выходные данные

Расширение в нижнем регистре либо NO.

Пример 1

Ввод · запись строки

```
'archive.tar.GZ'
```

Вывод · запись текста

```
'gz'
```

Пример 2

Ввод · запись строки

```
'.env'
```

Вывод · запись текста

```
'NO'
```

Подсказка / самопроверка

Нужна последняя, а не первая точка. Проверь обе границы и только затем возьми срез справа от точки.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.

D14. Команда учебного бота

Файл: `d14.py` · Дополнительно

Вводится команда, возможно с аргументами после пробелов. Выдели первое слово по пробельным промежуткам и сравни его без учёта регистра. Для /start выведи START, для /stop — STOP, для любого другого слова либо пустого ввода — UNKNOWN. /starter не является /start.

Выходные данные

Одна строка: START, STOP или UNKNOWN.

Пример 1

Ввод · запись строки

```
'  /START робот'
```

Вывод · запись текста

```
'START'
```

Пример 2

Ввод · запись строки

```
'/starter'
```

Вывод · запись текста

```
'UNKNOWN'
```

Подсказка / самопроверка

Перед parts[0] проверь, что список не пуст. Сравни выделенное слово целиком.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.

D15. Очищенный список тегов

Файл: `d15.py` · Дополнительно

Вводится строка тегов, разделённых запятыми. Для каждого поля убери пробельные края, переведи буквы в нижний регистр и выведи непустые теги по одному на строке. Порядок и повторы сохрани; внутренние пробелы не меняй. Если непустых тегов нет, выведи EMPTY.

Выходные данные

Обработанные непустые теги, каждый на новой строке, либо EMPTY.

Пример 1

Ввод · запись строки

```
' Python, , РОБОТ,python,'
```

Вывод · запись текста

```
'python\nробот\npython'
```

Пример 2

Ввод · запись строки

```
' , , '
```

Вывод · запись текста

```
'EMPTY'
```

Подсказка / самопроверка

Как в D11, обрабатывай поля по одному. Счётчик поможет узнать, был ли напечатан хотя бы один тег.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.

D16. Анализ строки без пробельных символов

Файл: `d16.py` · Дополнительно

Вводится строка. Удали из неё все пробельные символы, соединив элементы split() пустой строкой, и переведи буквы в нижний регистр. Если результат пустой, выведи EMPTY. Иначе проверь, читается ли он одинаково слева направо и справа налево. Знаки препинания сохраняются и участвуют в сравнении.

Выходные данные

YES, если непустой результат симметричен, NO в остальных непустых случаях, EMPTY для пустого результата.

Пример 1

Ввод · запись строки

```
'А роза упала на лапу Азора'
```

Вывод · запись текста

```
'YES'
```

Пример 2

Ввод · запись строки

```
'робот'
```

Вывод · запись текста

```
'NO'
```

Подсказка / самопроверка

Сначала обработай строку. Срез s[::-1] возвращает её символы в обратном порядке; это дополнительный приём, не метод.

Проверь себя: придумай ещё два входа, заранее запиши ожидаемые ответы и сравни с программой. Объясни, какие методы ты выбрал.
