ОТДЕЛЬНЫЙ УЧЕБНЫЙ КОМПЛЕКТ · PYTHON 3

Памятка по строковым методам

Что выбрать · что вернётся · какие ошибки проверить

Основные методы

Не учи всю таблицу за один раз. Сначала определи задачу и тип нужного результата. Квадратные скобки в сигнатуре означают необязательный аргумент, при вызове их писать не нужно.

Метод	Назначение	Пример	Результат
`lower()`	Нижний регистр	`"Py".lower()`	`"py"`
`upper()`	Верхний регистр	`"Py".upper()`	`"PY"`
`capitalize()`	Первый символ в верхний регистр, остальные в нижний	`"пРИВЕТ".capitalize()`	`"Привет"`
`strip()`	Пробельные края с обеих сторон	`" a b ".strip()`	`"a b"`
`lstrip()`	Пробельные символы слева	`" a ".lstrip()`	`"a "`
`rstrip()`	Пробельные символы справа	`" a ".rstrip()`	`" a"`
`replace(old, new[, n])`	Замена всех / первых n фрагментов	`"a-a".replace("a", "b", 1)`	`"b-a"`
`find(part)`	Индекс первого вхождения, иначе −1	`"robot".find("o")`	`1`
`rfind(part)`	Индекс последнего вхождения, иначе −1	`"robot".rfind("o")`	`3`
`index(part)`	Индекс первого вхождения; при отсутствии ValueError	`"robot".index("r")`	`0`
`count(part)`	Число неперекрывающихся вхождений	`"aaaa".count("aa")`	`2`
`startswith(prefix)`	Проверка начала	`"/go bot".startswith("/go")`	`True`
`endswith(suffix)`	Проверка конца	`"x.py".endswith(".py")`	`True`
`isalpha()`	Непустая строка из букв	`"Робот".isalpha()`	`True`
`isdigit()`	Непустая строка из цифр Unicode	`"007".isdigit()`	`True`
`isalnum()`	Непустая строка из букв и/или цифр	`"bot7".isalnum()`	`True`
`isspace()`	Непустая строка из пробельных символов	`"  ".isspace()`	`True`
`split()`	Список слов по пробельным промежуткам	`" a  b ".split()`	`["a", "b"]`
`split(sep[, n])`	Список полей по заданному разделителю	`"a;;b".split(";")`	`["a", "", "b"]`
`join(parts)`	Сборка строк через разделитель	`"-".join(["a", "b"])`	`"a-b"`

Дополнительно: по необходимости

Эти методы не нужны для обязательного маршрута. Сначала освой основные; сюда возвращайся, когда появится соответствующая задача.

Метод	Назначение	Пример	Результат
`title()`	Начальные буквы слов в верхний регистр; остальные в нижний	`"клуб РОБОТОВ".title()`	`"Клуб Роботов"`
`casefold()`	Более полная нормализация регистра Unicode, чем lower	`"Straße".casefold()`	`"strasse"`
`removeprefix(p)`	Удаление точного префикса один раз; Python 3.9+	`"/start".removeprefix("/")`	`"start"`
`removesuffix(p)`	Удаление точного суффикса один раз; Python 3.9+	`"happy.py".removesuffix(".py")`	`"happy"`
`zfill(n)`	Дополнение строки нулями слева до ширины n	`"72".zfill(4)`	`"0072"`
`isascii()`	Все символы ASCII; для пустой строки True	`"007".isascii()`	`True`

> `title()` использует свои правила границ слов и может неожиданно менять имена или названия. `zfill()` дополняет, но не проверяет цифры. `isascii()` сам по себе разрешает и буквы, и знаки: для ASCII-цифр нужно `s.isascii() and s.isdigit()`.

Пять правил

Правило	Пример
Сохраняй результат преобразования	`s = s.strip()`
Для наличия используй in	`"@" in s`
Не путай strip и replace	Края / вхождения в любом месте
split возвращает список	`words = s.split()`
join вызывает строка-разделитель	`" ".join(words)`

Длина — `len(s)`; символ — `s[i]`; срез — `s[a:b]`; переворот — `s[::-1]`; наличие — `part in s`. Это полезные операции, но не методы строк.

Источники и границы темы

Справочник для сверки поведения: официальная документация Python: строковые методы. Учебный контекст: твой курс Python для 7 класса, темы ввода, условий и циклов. Просмотрено 05.11.2026.

Материал и задачи подготовлены отдельно для этого курса. Авторы примеров не требуют внешних библиотек, функций, регулярных выражений или полного курса по спискам. Базовые решения подходят для Python 3. Дополнительные removeprefix/removesuffix требуют Python 3.9+.
