

Python: Условия внутри тела цикла

В теле цикла, как и в теле функции, можно выполнять инструкции. Поэтому внутри цикла можно использовать всё изученное ранее, например — условные конструкции.

Представьте функцию, которая считает, сколько раз входит буква в предложение. Пример ее работы:

```
count_chars('Fear cuts deeper than swords.', 'e') # 4
# Если вы ничего не нашли, то результат — 0 совпадений
count_chars('Sansa', 'y') # 0
```

Перед тем как посмотреть содержимое функции, подумайте:

- Является ли эта операция агрегацией?
- Какой будет проверка на вхождение символа?

```
def count_chars(string, char):
    index = 0
    count = 0
    while index < len(string):
        if string[index] == char:
            # Считаем только подходящие символы
            count = count + 1
        # Счетчик увеличивается в любом случае
```

```
    index = index + 1  
    return count
```

<https://replit.com/@hexlet/python-basics-loops-conditions-inside-loops>

Это агрегирующая задача. Несмотря на то, что она считает не все символы, чтобы подсчитать сумму, приходится анализировать каждый символ. Ключевое отличие этого цикла от рассмотренных — внутри тела есть условие.

Переменная `count` увеличивается только в том случае, когда рассматриваемый символ совпадает с ожидаемым. В остальном — это типичная агрегатная функция, которая возвращает количество нужных символов.

Задание

Функция из теории учитывает регистр букв. То есть `a` и `A` с её точки зрения разные символы. Реализуйте вариант этой же функции, так чтобы регистр букв был не важен:

```
count_chars('HexlEt', 'e') # 2  
count_chars('HexlEt', 'E') # 2
```

► Упражнение не проходит проверку — что делать?



- ▶ В моей среде код работает, а здесь нет 🤔
- ▶ Мой код отличается от решения учителя 🤔
- ▶ Прочитал урок — ничего не понятно 🙄