Python: Возврат из циклов

Работа с циклами обычно сводится к двум сценариям:

1. Агрегация. Накопление результата во время итераций и работа с ним после цикла. Переворот строки относится к такому варианту
2. Выполнение цикла до достижения необходимого результата и выход. Например, задача поиска простых чисел — которые делятся без остатка только на себя и на единицу

Рассмотрим алгоритм проверки простоты числа. Будем делить искомое число `x` на все числа из диапазона от двух до `x - 1` и смотреть остаток. Если в этом диапазоне не найден делитель, который делит число `x` без остатка, значит, перед нами простое число.

В этом случае достаточно проверять числа не до `x - 1`, а до половины числа. Например, 11 не делится на 2, 3, 4, 5. Но и дальше не будет делиться на числа больше своей половины. Значит, можно оптимизировать алгоритм и проверять деление только до `x / 2`:

```
def is_prime(number):
    if number < 2:
        return False
```

```
    divider = 2

    while divider <= number / 2:
        if number % divider == 0:
            return False

        divider += 1

    return True

print(is_prime(1))  # => False
print(is_prime(2))  # => True
print(is_prime(3))  # => True
print(is_prime(4))  # => False
```

https://replit.com/@hexlet/python-basics-loops-return-from-loops

Представим, что по алгоритму последовательного деления на числа до `x / 2` нашлось одно, которое делит без остатка. Значит, переданный аргумент — не простое число, и дальнейшие вычисления не имеют смысла. В этом месте стоит возврат `False`.

Если цикл отработал целиком, и не нашлось число, которое делит без остатка, значит, число — простое.

Задание

Реализуйте функцию `is_contains_char()`, которая проверяет с учётом регистра, содержит ли строка

указанную букву. Функция принимает два параметра:

- Строка
- Буква для поиска

```
print(is_contains_char('Hexlet', 'H'))  # => True
print(is_contains_char('Hexlet', 'h'))  # => False
print(is_contains_char('Awesomeness', 'm'))  # => True
print(is_contains_char('Awesomeness', 'd'))  # => False
```

▶ 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 🙄

Полезное

- Список простых чисел

Определения

- Агрегация — Накопление результата во время итераций и работа с ним после цикла.