

Python: Отрицание

Наряду с логическими операторами **И** и **ИЛИ**, часто используется операция «**отрицание**». Она меняет логическое значение на противоположное. В программировании отрицанию соответствует унарный оператор `not`:

```
not True    # False
not False   # True
```

Например, если есть функция, которая проверяет четность числа, то с помощью отрицания можно выполнить проверку нечетности:

```
def is_even(number):
    return number % 2 == 0

print(is_even(10))      # => True
print(not is_even(10))  # => False
```

В примере выше мы добавили `not` слева от вызова функции и получили обратное действие.

Отрицание — инструмент, с которым можно выражать задуманные правила в коде и не писать новые функции.

Если написать `not not is_even(10)`, то код сработает

даже в таком случае:

```
print(not not is_even(10))  # => True
```

<https://replit.com/@hexlet/python-basics-logic-logical-negation>

В логике двойное отрицание — это отсутствие отрицания:

```
not not True    # True
```

```
not not False   # False
```

```
print(not not is_even(10))  # => True
```

```
print(not not is_even(11))  # => False
```

Теперь вы знаете, что означают операторы **И**, **ИЛИ** и **not**. С их помощью вы сможете задавать составные условия из двух и более логических выражений.

Задание

1. Реализуйте функцию `is_palindrome()`, которая определяет, является ли слово палиндромом или нет. Палиндром - это слово, которое читается одинаково в обоих направлениях. Слова в функцию могут быть переданы в любом регистре, поэтому сначала нужно привести слово к нижнему регистру: `word.lower()`.

```
is_palindrome('шалаш') # true
is_palindrome('хекслет') # false
is_palindrome('Довод') # true
is_palindrome('Функция') # false
```

2. Реализуйте функцию `is_not_palindrome()`, которая проверяет что слово НЕ является палиндромом:

```
is_not_palindrome('шалаш') # false
is_not_palindrome('Ага') # false
is_not_palindrome('хекслет') # true
```

Для этого, вызовите функцию `is_palindrome()` внутри `is_not_palindrome()` и примените отрицание.

► 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

- [Законы Де Моргана](#)

Определения

- Отрицание — логическая операция, которая

меняет логическое значение на
противоположное.