

Python: Тернарный оператор

Посмотрите на определение функции, которая возвращает модуль переданного числа:

```
def abs(number):  
    if number >= 0:  
        return number  
    return -number
```

Но можно записать более лаконично. Для этого справа от `return` должно быть выражение, но `if` — это инструкция, а не выражение. В Python есть конструкция, которая работает как `if-else`, но считается выражением. Она называется **тернарный оператор** — единственный оператор в Python, который требует три операнда:

```
def abs(number):  
    return number if number >= 0 else -number
```

Общий паттерн выглядит так: `<expression on true> if <predicate> else <expression on false>`.

Давайте перепишем начальный вариант `get_type_of_sentence()` аналогично.

Было:

```
def get_type_of_sentence(sentence):  
    last_char = sentence[-1]  
    if last_char == '?':  
        return 'question'  
    return 'normal'
```

Стало:

```
def get_type_of_sentence(sentence):  
    last_char = sentence[-1]  
    return 'question' if last_char == '?' else 'normal'  
  
print(get_type_of_sentence('Hodor'))    # => normal  
print(get_type_of_sentence('Hodor?'))   # => question
```

<https://replit.com/@hexlet/python-basics-conditionals-ternary-operator>

Тернарный оператор можно вкладывать в тернарный оператор. Но не нужно так делать, так как такой код тяжело читать и отлаживать.

Задание

Реализуйте функцию `flip_flop()`, которая принимает на вход строку и, если эта строка равна `'flip'`, возвращает строку `'flop'`. В противном случае функция должна вернуть `'flip'`.

Примеры вызова:

```
print(flip_flop('flip')) # => 'flop'  
print(flip_flop('flop')) # => 'flip'
```

Попробуйте написать два варианта функции: с обычным if-else, и с тернарным оператором.

► 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 🤔

Определения

- Тернарный оператор — способ превратить простую условную инструкцию в выражение, например, `number if number >= 0 else -number`.