

Python: Конструкция else + if = elif

Функция `get_type_of_sentence()` различает только вопросительные и обычные предложения. Добавим в нее поддержку восклицательных предложений:

```
def get_type_of_sentence(sentence):
    last_char = sentence[-1]

    if last_char == '?':
        sentence_type = 'question'

    if last_char == '!':
        sentence_type = 'exclamation'
    else:
        sentence_type = 'normal'

    return 'Sentence is ' + sentence_type

print(get_type_of_sentence('Who?')) # => 'Sentence is norm
print(get_type_of_sentence('No'))   # => 'Sentence is norm
print(get_type_of_sentence('No!'))  # => 'Sentence is excl
```

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

Мы добавили проверку восклицательных предложений — `exclamation`. Технически эта функция работает, но вопросительные предложения трактует неверно. Еще в ней есть проблемы с точки зрения

семантики:

- Наличие восклицательного знака проверяется в любом случае, даже если уже обнаружился вопросительный знак
- Ветка `else` описана для второго условия, но не для первого. Поэтому вопросительное предложение становится `"normal"`

Чтобы исправить ситуацию, воспользуемся еще одной возможностью условной конструкции:

```
def get_type_of_sentence(sentence):
    last_char = sentence[-1]

    if last_char == '?':
        sentence_type = 'question'
    elif last_char == '!':
        sentence_type = 'exclamation'
    else:
        sentence_type = 'normal'

    return 'Sentence is ' + sentence_type

print(get_type_of_sentence('Who?')) # => 'Sentence is ques
print(get_type_of_sentence('No'))  # => 'Sentence is norm
print(get_type_of_sentence('No!')) # => 'Sentence is excl
```

Теперь все условия выстроились в единую конструкцию. `elif` означает — «если не выполнено предыдущее условие, но выполнено текущее».

Получается такая схема:

- если последняя буква `?`, то `'question'`
- если последняя буква `!`, то `'exclamation'`
- остальные варианты — `'normal'`

Выполнится только один из блоков кода, который относится ко всей конструкции `if`.

Задание

На электронной карте Вестероса, которую реализовал Сэм, союзники Старков отображены зелёным кружком, враги — красным, а нейтральные семьи — серым.

Напишите для Сэма функцию

`who_is_this_house_to_starks()`, которая принимает на вход фамилию семьи и возвращает одно из трёх значений: `'friend'`, `'enemy'`, `'neutral'`.

Правила определения:

- Друзья (`'friend'`): `'Karstark'`, `'Tully'`
- Враги (`'enemy'`): `'Lannister'`, `'Frey'`
- Любые другие семьи считаются нейтральными (`'neutral'`)

Примеры вызова:

```
print(who_is_this_house_to_starks('Karstark')) # => 'frien
print(who_is_this_house_to_starks('Frey'))      # => 'enemy
print(who_is_this_house_to_starks('Joar'))       # => 'neutr
print(who_is_this_house_to_starks('Ivanov'))     # => 'neutr
```

► 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

- `else + if = elif` — способ задать несколько альтернативных условий.