

# Python: Условная конструкция else

Теперь изменим функцию из предыдущего урока так, чтобы она возвращала не просто тип предложения, а целую строку `Sentence is normal` ИЛИ `Sentence is question`:

```
def get_type_of_sentence(sentence):  
    last_char = sentence[-1]  
  
    if last_char == '?':  
        sentence_type = 'question'  
    else:  
        sentence_type = 'normal'  
  
    return "Sentence is " + sentence_type  
  
print(get_type_of_sentence('Hodor'))    # => 'Sentence is no  
print(get_type_of_sentence('Hodor?'))  # => 'Sentence is qu
```

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

Мы добавили `else` и новый блок. Он выполнится, если условие в `if` — ложь. Еще в блок `else` можно вкладывать другие условия `if`. `Else` переводится «иначе», «в ином случае».

Оформить конструкцию `if-else` можно двумя способами. С помощью отрицания можно изменить порядок блоков:

```
def get_type_of_sentence(sentence):  
    last_char = sentence[-1]  
  
    if last_char != '?':  
        sentence_type = 'normal'  
    else:  
        sentence_type = 'question'  
  
    return "Sentence is " + sentence_type
```

Чтобы конструкцию было легче оформлять, старайтесь выбирать проверку без отрицаний и подстраивайте содержимое блоков под нее.

## Задание

Реализуйте функцию `normalize_url()`, которая выполняет нормализацию данных. Она принимает адрес сайта и возвращает его с `https://` в начале.

Функция принимает адреса в виде АДРЕС или `http://АДРЕС`, но всегда возвращает адрес в виде `https://АДРЕС`. На вход функции также может поступить адрес в уже нормализованном виде `https://АДРЕС`, в этом случае ничего менять не надо.

## Примеры вызова:

```
print(normalize_url('https://ya.ru')) # => 'https://ya.ru'
print(normalize_url('google.com'))    # => 'https://google
print(normalize_url('http://ai.fi'))  # => 'https://ai.fi'
```

Есть несколько способов решить задачу. Один из них — сравнивать первые 7 символов строки-аргумента со строкой `http://`, а потом на основе этого добавлять или не добавлять к ней `https://`.

Также вам скорее всего потребуется отбросить ненужную часть в начале строки. Помните, мы рассматривали способ получения кусочка от строки с помощью среза? Если нет, напоминаю:

```
# Берём 6 символов от начала
print('Winterfell'[:6]) # => 'Winter'
```

Так вот, с помощью срезов можно также отбросить определённое количество символов:

```
# Отбрасываем первые 6 символов
print('Winterfell'[6:]) # => 'fell'
```

► 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` — способ задать блок кода, который будет выполнен, если условие с `if` не удовлетворено.