

# Python: Пограничные случаи

Функция `my_substr()`, которую вы реализовали в прошлом уроке, содержит много ошибок. Она прошла проверку, так как в ней не было **пограничных случаев**. Функция работала с нормальными аргументами. А теперь представим, что ей передали такие варианты длины:

- 0
- Отрицательное число
- Число, которое превышает реальный размер строки

Функция `my_substr()` не рассчитана на такие варианты. Код будет запускаться в разных ситуациях, с разными комбинациями условий и данных. Нельзя быть уверенным, что аргументы всегда будут корректными, поэтому нужно учитывать все случаи.

Ошибки в пограничных случаях — частая причина логических ошибок в программах. Программисты всегда забывают что-нибудь учесть. Такие ошибки часто проявляются не сразу и могут долго не приводить к видимым проблемам.

Программа продолжает работать, но в какой-то момент обнаруживается, что в результатах есть

ошибки. Часто причина в динамической типизации Python.

Вы научитесь справляться с такими ошибками с опытом.

Представим расширенную функцию `my_substr()`. Она принимает три аргумента: строку, индекс и длину извлекаемой подстроки. Функция возвращает подстроку указанной длины, начиная с указанного индекса. Примеры вызова:

```
string = 'If I look back I am lost'
print(my_substr(string, 0, 1))  # => 'I'
print(my_substr(string, 3, 6))  # => 'I look'
```

Какие **пограничные случаи** стоит учитывать:

- Отрицательная длина извлекаемой подстроки
- Отрицательный заданный индекс
- Заданный индекс выходит за границу всей строки
- Длина подстроки в сумме с заданным индексом выходит за границу всей строки

Когда функция реализуется, каждый пограничный случай будет отдельным куском кода. Скорее всего, он будет реализовываться с помощью `if`.

Чтобы написать функцию `my_substr()` и защититься от этих случаев, стоит реализовать отдельную функцию,

которая будет проверять аргументы на корректность.

## Задание

Реализуйте функцию-предикат

`is_arguments_for_substr_correct()`, которая принимает три аргумента:

1. строку;
2. индекс, с которого начинать извлечение;
3. длину извлекаемой подстроки.

Функция возвращает `False`, если хотя бы одно из условий истинно:

- Отрицательная длина извлекаемой подстроки.
- Отрицательный заданный индекс.
- Заданный индекс выходит за границу всей строки.
- Длина подстроки в сумме с заданным индексом выходит за границу всей строки.

В ином случае функция возвращает `True`.

Не забывайте, что индексы начинаются с 0, поэтому индекс последнего элемента — это «длина строки минус 1».

Пример вызова:

```
string = 'Sansa Stark'
```

```
print(is_arguments_for_substr_correct(string, 2, -3))    # =
print(is_arguments_for_substr_correct(string, -1, 3))    # =
print(is_arguments_for_substr_correct(string, 4, 100))   # =
print(is_arguments_for_substr_correct(string, 10, 10))   # =
print(is_arguments_for_substr_correct(string, 11, 1))    # =
print(is_arguments_for_substr_correct(string, 3, 3))     # =
print(is_arguments_for_substr_correct(string, 0, 5))     # =
```

► 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 🤔