

Замыкания

Задача 1. Используя замыкания функций, определите вложенную функцию, которая бы увеличивала значение переданного параметра на 5 и возвращала бы вычисленный результат. При этом внешняя функция должна иметь следующую сигнатуру:

```
def counter_add(): ...
```

Вызовите функцию `counter_add` и результат ее работы присвойте переменной с именем `cnt`. Вызовите внутреннюю функцию через переменную `cnt` со значением `k`, введенным с клавиатуры:

```
k = int(input())
```

Выведите результат на экран.

Задача 2. Используя замыкания функций, объявите внутреннюю функцию, которая увеличивает значение своего аргумента на некоторую величину `n` - параметр внешней функции с сигнатурой:

```
def counter_add(n): ...
```

Вызовите внешнюю функцию `counter_add` со значением аргумента 2 и результат присвойте переменной `cnt`. Вызовите внутреннюю функцию через переменную `cnt` со значением `k`, введенным с клавиатуры:

```
k = int(input())
```

Выведите результат на экран.

Задача 3. Используя замыкания функций, объявите внутреннюю функцию, которая заключает в тег `h1` строку `s` (`s` - строка, параметр внутренней функции). Далее, на вход программы поступает строка и ее нужно поместить в тег `h1` с помощью реализованного замыкания. Результат выведите на экран.

P. S. Пример добавления тега `h1` к строке "Python":

Python

Sample Input:

Hello World

Sample Output:

```
<h1>Hello World</h1>
```