**05.03.2019**

**Урок 4**

Рекурсия

РЕКУРСИЯ = РАЗДЕЛЯЙ И ВЛАСТВУЙ (DIVIDE AND CONQUER)



**Пример:**

def countdown(i):

print i

if i <= 0 Основное условие (Base condition)

return

else:

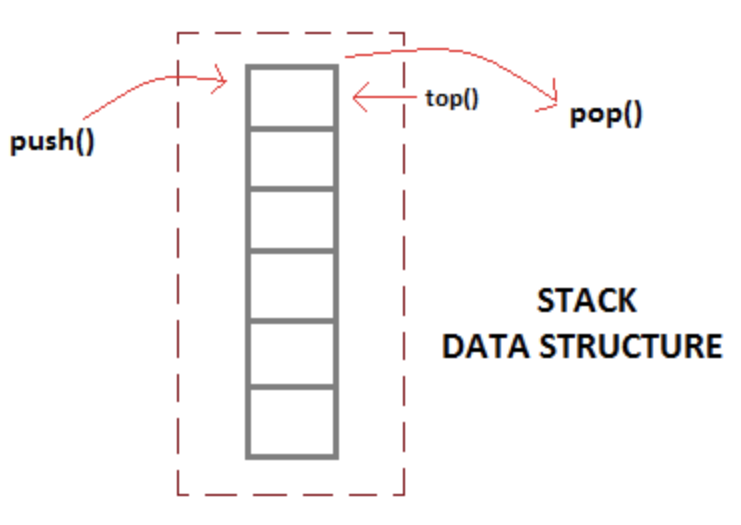
countdown(i-1) Рекурсивное условие (Recursive

condition)

**Рекурсия** – функция которая вызывает себя до тех пор, пока не исполнится **Основное условие**.

Пока выполняется рекурсивное условие, контекст выполнения будет размещаться на верхушку **стека** (stack). Это будет продолжаться до тех пор, пока не произойдет переполнение стека (stack overflow).

**Стек** – абстрактный тип данных представляющий собой список элементов.



Метод Разделяй и Властвуй чаще всего применяется в – двоичных деревьях, графах, heap, быстрая сортировка (quicksort) и heapsort (пирамидальная сортировка).

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| * Компактный и удобный для чтения код * Возможность разделить проблему на небольшие куски и решать по отдельности | * Требует больше памяти * Во многих языках программирования итерация занимает меньше времени |

Быстрая сортировка (Quicksort)

1. Выбрать любой элемент из массива. Этот элемент называется  пивот (pivot)
2. Найти элементы меньше пивота и больше пивота
3. Вызвать быструю сортировку рекурсивно для оставшихся массивов

|  |  |  |
| --- | --- | --- |
| 33 | 15 | 10 |

35

Цифры больше 33 (пустой массив)

пивот

цифры меньше 33

35

|  |  |
| --- | --- |
| 15 | 10 |

35

|  |  |
| --- | --- |
| 15 | 10 |

  Quicksort quicksort

35

10

15

|  |  |  |
| --- | --- | --- |
| 10 | 15 | 35 |

Sorted array

Def getFactorial(n)

If n<0:

Return -1

Elif n<2:

Return 1

Else:

Return n\*getFactorial(n-1)

Print(getFactorial())