Работа с рекурсией

№ урока:	5	**Курс:**	Python Базовый

Средства обучения:	Персональный компьютер/ноутбук стандартной производительности

Обзор, цель и назначение урока

Познакомиться с понятием рекурсии. На уроке будет рассмотрено как применять рекурсию на практических задачах, для чего она нужна и какие важные моменты нужно знать при работе с ней.

Изучив материал данного занятия, учащийся сможет:

- Понимать, что такое рекурсия и зачем она нужна.
- Уметь решать задачи, заменяя стандартные циклы рекурсией для оптимизации процессов.

Содержание урока

1. Что такое рекурсия
2. Как ее применить на практике
3. Решение задач

Резюме

- Рекурсия, по сути, это процесс, когда функция вызывает саму себя. Если у вас есть возможность поставить 2 зеркала друг перед другом и заглянуть в них, то сделайте это. Это и будет рекурсия в реальном мире.
- Только в отличие от реального мира рекурсия в программировании имеет свои особенности. Если вы напишите функцию def test_recursion: test_recursion(), то у вас вылетит ошибка Stack overflow (ошибка переполнения стека или Maximum recursion deph, максимальная глубина рекурсии).
- Поэтому, чтобы освоить рекурсию, нужно понимать, что в функции, помимо вызова самой себя, еще должны присутствовать 2 вещи: изменение состояния и условие выхода из рекурсии.
- Изменение состояния - абстрактное понятие. Если вы создадите переменную x=5 и передадите ее в функцию, выведете ее на экран командой print(x), после чего вызовите эту же функцию внутри ее (рекурсия), куда передадите уже x-1, то это будет изменение состояния.
- Условие же выхода, это если вы допишите if x == 0: return и рекурсия прервется. Таким образом вы выведете на экран цифры 5,4,3,2,1,0 не используя ни одного цикла.
- У рекурсии есть свои достоинства и недостатки. Достоинством является скорость выполнения программы. Рекурсия гораздо быстрее циклов. Однако из недостатков можно выделить сложность ее понимания (понять рекурсию сложнее чем понять циклы), а с технической стороны у рекурсии есть лимит вызовов. В Python он равен 1000. То есть если вы хотите работать рекурсивно со структурами данных с длиной более 1000, то у вас все сломается. Так что вы можете на более высоком уровне определить, что

CyberBionic systematics

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

если длина списка меньше 1000, то можно использовать рекурсивную функцию для скорости, а если больше, то цикл.
- Лимит служит защитой от полного краха программы и зависания системы. Если убрать лимит, рекурсия заполнит весь стек вызовов, тем самым “съест” всю оперативную память и вам поможет только грубый перезапуск компьютера кнопкой.
- В случае с бесконечным циклом такого не будет, программа просто будет работать пока включен компьютер или ее не прервут из вне. При этом другие программы не будут затронуты.

Закрепление материала

- Что такое рекурсия?
- Зачем нужна рекурсия? Чем она лучше циклов?
- Чем она хуже циклов?
- Что будет, если забыть прописать условие выхода в рекурсивной функции?

Дополнительное задание

Зависит ли максимальная глубина рекурсии от вычислительной мощности компьютера? Зависит ли она от размера данных, которые изменяются внутри рекурсии?
Выясните эти моменты. Изучите теорию рекурсии глубже.

Самостоятельная деятельность учащегося

1. Напишите рекурсивную функцию, чтобы сгенерировать и вернуть список от 1 до N. Аргументом функции является только N.
2. Напишите функцию, которая рекурсивно ищет в сложном объекте значение. Сложный объект — это будет список списков списков и т.д. Например, [1, 2, [3, 4, [5, [6, []]]]]. Функция должна рекурсивно заходить внутрь вложенных массивов, а если это другой тип данных игнорировать его.

Рекомендуемые ресурсы

- Recursion (w3schools)
- https://pythonru.com/osnovy/rekursiya-python
- https://codecamp.ru/blog/python-recursion/
- Recursion

CyberBionic systematics

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com