

Робота з рекурсією

№ уроку: 5 **Курс:** Python Базовий

Засоби навчання: Персональний комп'ютер/ноутбук стандартної продуктивності

Огляд, мета та призначення уроку

Познайомитись із поняттям рекурсії. На уроці буде розглянуто, як застосовувати рекурсію на практичних завданнях, для чого вона потрібна і які важливі моменти потрібно знати під час роботи з нею.

Вивчивши матеріал даного заняття, учень зможе:

- Розуміти, що таке рекурсія та навіщо вона потрібна.
- Вміти вирішувати завдання, замінюючи стандартні цикли рекурсією для оптимізації процесів.

Зміст уроку

1. Що таке рекурсія
2. Як її застосувати на практиці
3. Вирішення задач

Резюме

- Рекурсія, власне, це процес, коли функція викликає себе. Якщо ви маєте можливість поставити 2 дзеркала один перед одним і заглянути в них, то зробіть це. Це і буде рекурсія у реальному світі.
- Тільки на відміну реального світу рекурсія у програмуванні має свої особливості. Якщо ви напишіть функцію `def test_recursion: test_recursion()`, то у вас вилетить помилка `Stack overflow` (помилка переповнення стека або `Maximum recursion depth`, максимальна глибина рекурсії).
- Тому, щоб освоїти рекурсію, треба розуміти, що у функції, крім виклику самої себе, ще повинні бути присутніми дві речі: зміна стану та умова виходу з рекурсії.
- Зміна стану – абстрактне поняття. Якщо ви створите змінну `x=5` і передасте її в функцію, виведете її на екран командою `print(x)`, після чого викличте цю функцію всередині її (рекурсія), куди передасте вже `x-1`, це буде зміна стану.
- Умова ж виходу, якщо ви допишете `if x == 0: return` і рекурсія перерветься. Таким чином, ви виведете на екран цифри 5,4,3,2,1,0 не використовуючи жодного циклу.
- Рекурсія має свої переваги та недоліки. Перевагою є швидкість виконання програми. Рекурсія набагато швидше за цикли. Однак із недоліків можна виділити складність її розуміння (зрозуміти рекурсію складніше ніж зрозуміти цикли), а з технічного боку рекурсія має ліміт викликів. У Python він дорівнює 1000. Тобто, якщо ви хочете працювати рекурсивно зі структурами даних з довжиною більше 1000, то у вас все зламається. Так що ви можете на вищому рівні визначити, що якщо довжина списку менше 1000, то можна використовувати рекурсивну функцію для швидкості, а якщо більше, то цикл.

- Ліміт служить захистом від повного краху програми та зависання системи. Якщо прибрати ліміт, рекурсія заповнить весь стек викликів, тим самим з'їсть всю оперативну пам'ять і вам допоможе лише грубий перезапуск комп'ютера кнопкою.
- У випадку з нескінченним циклом такого не буде, програма просто працюватиме поки включений комп'ютер або її не перервуть з зовні. При цьому інші програми не будуть зачеплені.

Закріплення матеріалу

- Що таке рекурсія?
- Навіщо потрібна рекурсія? Чим вона краща за цикли?
- Чим вона гірша за цикли?
- Що буде, якщо забути прописати умову виходу до рекурсивної функції?

Додаткове завдання

Чи залежить максимальна глибина рекурсії від обчислювальної потужності комп'ютера? Чи залежить вона від обсягу даних, які змінюються всередині рекурсії? З'ясуйте ці моменти. Вивчіть теорію рекурсії глибше.

Самостійна діяльність учня

1. Напишіть рекурсивну функцію, щоб згенерувати та повернути список від 1 до N. Аргументом функції є тільки N.
2. Напишіть функцію, яка рекурсивно шукає у складному об'єкті значення. Складний об'єкт – це буде список списків списків і т.д. Наприклад, [1, 2, [3, 4, [5, [6, []]]]]. Функція має рекурсивно заходити всередину вкладених масивів, і якщо це інший тип даних ігнорувати його.

Рекомендовані ресурси

- [Recursion \(w3schools\)](https://www.w3schools.com/python/python_recursion.asp)
- <https://pythonru.com/osnovy/rekursiya-python>
- <https://codecademy.com/learn/python-recurstion/>
- [Recursion](https://www.it-ebooks.info/book/1000000/recursion)