**Что такое рекурсия?**

Рекурсия - это термин в программировании, означающий вызов функцией самой себя.

Это метод решения задач, похожий на математическую индукцию: чтобы функция выполнилась, нужно сначала получить ее результат при вызове с другим значением.

**Что такое шаг рекурсии?**

шаг рекурсии, он же условие продолжения – способ сведения задачи к более простой.

**Что такое глубина рекурсии?**

Общее количество вложенных вызовов (включая первый) необходимый для достижения базы рекурсии.

**Что такое база рекурсии?**

База рекурсии - это такие аргументы функции, которые делают задачу настолько простой, что решение не требует дальнейших вложенных вызовов – это условие остановки рекурсии.

**На сколько большой может быть глубина рекурсии?**

Максимальная глубина рекурсии ограничена движком JavaScript. Точно можно рассчитывать на 10000 вложенных вызовов, некоторые интерпретаторы допускают и больше, но для большинства из них 100000 вызовов - за пределами возможностей.

**Почему рекурсия энергозатратна?**

Рекурсия приводит к хранению всех контекстов для неоконченных внешних вызовов в стеке. Для простой рекурсии реализация через цикл гораздо более экономна, потому цикл будет иметь один контекст.

**Как связаны рекурсия и цикл?**

Любая рекурсия может быть переделана в цикл, т.е. любая рекурсивная функция может быть переписана в итеративную.

Как правило, вариант с циклом будет эффективнее. И это иногда требуется для оптимизации работы.

**Какая структура данных называется рекурсивной (рекурсивно определяемая) структура данных? Например?**

Это структура, которая повторяет саму себя в своих частях. Например, связный список, DOM (которое явдяется деревом) это структура, в которой у каждого узла может быть несколько дочерних подузлов — «детей».

**Когда удобно использовать рекурсию?**

Удобно использовать для обхода рекурсивных структур данных.

Для задач, которые можно разбить на подзадачи, которые представляют собой меньшие версии самой себя, например, вычисление факториала.

Если не важна скорость – то рекурсия.