1. Перечислите основные глобальные объекты Node.js и поясните их назначение.

global - хранит var-данные на уровне модуля

process - информация о среде выполнения, о текущем процессе

buffer - класс Buffer – предназначен для работы с двоичными данными.

console - используется для печати в stdout и stderr

1. Поясните понятие «асинхронная функция».

Асинхронная функция - это функция, которая выполняется в фоновом режиме, не блокируя основной поток выполнения программы. Она позволяет выполнять операции ввода-вывода, сетевые запросы и другие задачи, которые могут занимать время, без ожидания их завершения. Асинхронные функции обычно принимают обратные вызовы (callback) или возвращают объект Promise для обработки результата выполнения.

1. Поясните понятие «стандартные системные потоки».

Стандартные системные потоки - это три основных потока ввода и вывода, доступные в большинстве операционных систем:

stdin (стандартный вход): Поток для ввода данных, обычно с клавиатуры. В Node.js это представлено как process.stdin.

stdout (стандартный вывод): Поток для вывода данных, обычно на экран. В Node.js это представлено как process.stdout.

stderr (стандартный вывод ошибок): Поток для вывода сообщений об ошибках, обычно на экран. В Node.js это представлено как process.stderr.

1. Поясните назначение функций **process.nextTick**, **setImmediate**. Поясните в чем их разница.

В Node.js функции process.nextTick и setImmediate выполняются в разные моменты времени, и скорость их выполнения зависит от текущей очереди событий и контекста выполнения. Давайте разберемся, когда каждая из них выполнится и почему:

process.nextTick:

process.nextTick помещает функцию в начало текущей фазы событий (event loop).

Это означает, что функция будет выполнена сразу после завершения текущей операции I/O, но до любых других событий в очереди.

Это позволяет создавать "микрозадачи" (microtasks), которые выполняются максимально быстро и приоритетно перед другими событиями.

process.nextTick может вызываться рекурсивно, и это может привести к глубокому стеку вызовов, если злоупотреблять им.

setImmediate:

setImmediate помещает функцию в очередь событий, но в отличие от process.nextTick, она выполняется после текущей фазы событий.

Это означает, что, если есть другие задачи в очереди событий (например, таймеры), они будут выполнены перед setImmediate.

setImmediate обычно более эффективен, когда требуется выполнить код асинхронно, но не сразу после завершения текущей операции I/O.

Время выполнения зависит от контекста выполнения и того, что именно вы хотите достичь. Если вам нужно выполнить код как можно скорее, и это не зависит от текущей очереди событий, то process.nextTick будет выполняться быстрее, так как он имеет более высокий приоритет.