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

Вот некоторые из основных глобальных объектов Node.js:

Console - предоставляет способ записи информации на консоль.

Process - предоставляет информацию о текущем процессе Node.js и позволяет контролировать процесс.

Buffer - предоставляет возможность напрямую работать с двоичными данными.

Global - предоставляет способ определения глобальных переменных и функций, к которым можно получить доступ из любого места в коде.

Module - предоставляет способ определения и экспорта модулей из приложения Node.js.

2. Поясните понятие «асинхронная функция».

Асинхронная функция — это функция, предназначенная для неблокирующего выполнения. Когда вызывается асинхронная функция, она может запустить операцию, выполнение которой занимает много времени, например чтение из файла или выполнение сетевого запроса. Вместо того, чтобы ждать завершения операции перед возвратом значения, функция немедленно возвращает значение, позволяя продолжить выполнение другого кода. Когда операция завершится, функция выполнит обратный вызов или вернет обещание с результатом.

3. Поясните понятие стандартные «системные потоки».

Стандартные «системные потоки» относятся к потокам, управляемым ядром операционной системы. Эти потоки обычно используются для выполнения низкоуровневых задач операционной системы, таких как обработка сетевых запросов или управление операциями файловой системы. Node.js по умолчанию использует один поток для выполнения всего кода JavaScript. Однако он может использовать пул потоков операционной системы для выполнения операций ввода-вывода в отдельном потоке, не блокируя основной поток.

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

Функция process.nextTick() и функция setImmediate() используются для планирования запуска функции в будущем, но с немного другим поведением.

process.nextTick(): планирует выполнение функции обратного вызова сразу после завершения текущей функции, но до продолжения цикла обработки событий. Это означает, что любой ввод-вывод или другие события, происходящие в течение текущего цикла цикла событий, будут обработаны до выполнения обратного вызова. Это может быть полезно для разбивки длительных задач или выполнения работы, которую нужно выполнить как можно скорее.

setImmediate(): планирует выполнение функции обратного вызова в следующем цикле цикла обработки событий. Это означает, что любой ввод-вывод или другие события, происходящие в течение текущего цикла цикла событий, будут обработаны до выполнения обратного вызова. Это может быть полезно для выполнения работы, которая должна быть выполнена после завершения текущего цикла цикла событий.

Основное различие между этими двумя функциями заключается в том, что process.nextTick() планирует выполнение обратного вызова перед любыми другими событиями ввода-вывода, а setImmediate() планирует выполнение обратного вызова после любых других событий ввода-вывода.