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

Глобальный объект предоставляет переменные и функции, доступные в любом месте программы. По умолчанию это те, что встроены в язык или среду исполнения.

В браузере он называется window, в Node.js — global, в другой среде исполнения может называться иначе.

Объект ***process*** принадлежит к числу важнейших компонентов среды Node, так как он предоставляет информацию о среде выполнения. Кроме того, через объект process выполняется стандартный ввод/вывод, вы можете корректно завершить приложение Node и даже выдать сигнал при завершении итерации в цикле событий Node.

1. **Поясните понятие «асинхронная функция».**

**Понятие асинхронности**: операция называется асинхронной, если ее выполнение осуществляется в 2 фазы: 1) заявка на исполнение; 2) получение результата; при этом участвуют два механизма: A-механизм, формирующий заявку и потом получающий результат; B-механизм, получающий заявку от A, исполняющий операцию и отправляющий результат A; продолжительность исполнения операции B-механизмом, как правило, непредсказуемо; в то время пока B-механизм исполняет операцию, А-механизм выполняет собственную работу. Применение асинхронности не противоречит применению многопоточности.

Асинхронность позволяет вынести отдельные задачи из основного потока в специальные асинхронные методы или блоки кода.

1. **Поясните понятие стандартные «системные потоки».**

Поток – байтовая последовательность, передаваемая в процессе ввода-вывода.

Стандартные потоки ввода-вывода в [системах](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) типа [UNIX](https://ru.wikipedia.org/wiki/UNIX) (и некоторых других) — [потоки](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%BE%D0%BA_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) [процесса](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), имеющие номер (дескриптор), зарезервированный для выполнения некоторых «стандартных» функций

Stdin, stdout, stderr, stdprn, stdaux.

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

Для того, чтобы поставить функцию в очередь на выполнение без задержки, в Microsoft предложили метод [setImmediate(func)](http://msdn.microsoft.com/en-us/library/ie/hh773176.aspx). Он реализован в IE10+ и на платформе Node.JS.

У setImmediate единственный аргумент – это функция, выполнение которой нужно запланировать.

Метод process.nextTick() добавляет функцию обратного вызова в «очередь следующего периода». После того, как текущий оборот цикла событий подходит к завершению, все функции обратного вызова будут вызваны в очереди следующего периода.

Используйте setImmediate, если вы хотите поставить в очередь функцию за любыми обратными вызовами событий ввода-вывода, которые уже находятся в очереди событий. Используйте process.nextTick для эффективной очереди функции во главе очереди событий, чтобы она выполнялась сразу после завершения текущей функции.

Итак, в случае, когда вы пытаетесь расстаться с длинной работой, связанной с процессором, с использованием рекурсии, теперь вы хотите использовать setImmediate вместо process.nextTick для очередности следующей итерации, поскольку в противном случае любые операции ввода/Вывода не будет иметь шансов работать между итерациями.