**Глобальные объекты Node.js:**

* global: глобальный объект, доступный в любом модуле. Аналог window в браузере.
* process: объект, представляющий текущий процесс Node.js. Управляет средой, переменными окружения, потоками.
* \_\_dirname: абсолютный путь к директории текущего модуля.
* \_\_filename: абсолютный путь к файлу текущего модуля.
* module: объект, представляющий текущий модуль.
* exports: объект, с помощью которого экспортируются функции и данные модуля.
* Buffer: используется для работы с бинарными данными.

**19. Асинхронная функция**

Асинхронная функция — это функция, которая выполняет асинхронные операции, не блокируя основной поток выполнения. Она определяется с помощью ключевого слова async и может содержать оператор await, который позволяет "ожидать" завершения промисов. Асинхронные функции упрощают работу с асинхронным кодом, делая его более читаемым и управляемым.

**20. Стандартные системные потоки**

Стандартные системные потоки в Node.js — это встроенные потоки для ввода и вывода данных. Существует три основных потока:

stdin: Стандартный поток ввода, используемый для получения данных от пользователя или другой программы.

stdout: Стандартный поток вывода, используемый для вывода данных на консоль или в другой поток.

stderr: Стандартный поток ошибок, предназначенный для вывода сообщений об ошибках и предупреждений.

Эти потоки позволяют взаимодействовать с пользовательским вводом и выводом данных в консоль.

**21. Функции process.nextTick и setImmediate**

process.nextTick(callback): Эта функция добавляет callback в очередь, которая будет выполнена после текущей операции, но перед выполнением любых других операций в цикле событий. Это позволяет выполнять код, который должен быть выполнен сразу после завершения текущей операции, но перед любыми другими запланированными задачами.

setImmediate(callback): Эта функция добавляет callback в очередь, которая будет выполнена в следующем цикле событий, после завершения всех текущих операций и других запланированных задач. Она используется для выполнения кода после текущего цикла событий.

Разница: Основная разница между process.nextTick и setImmediate заключается в том, когда они выполняются. process.nextTick выполняется перед следующими операциями в цикле событий, а setImmediate — после. Это может повлиять на порядок выполнения кода, особенно в асинхронных сценариях.