1. Шаблонизация - это процесс создания и использования шаблонов, которые определяют структуру и внешний вид документа или данных. Она позволяет разделять логику и данные от представления, что облегчает разработку и поддержку приложений. Шаблонизация используется во многих областях, включая веб-разработку, генерацию документов, электронную почту и др.
2. Шаблон (template) - это файл или фрагмент кода, содержащий комбинацию статического текста и динамических элементов, которые могут быть заполнены данными во время выполнения. Шаблоны обычно содержат заполнители или переменные, которые заменяются конкретными значениями при рендеринге или исполнении.
3. Контекст шаблона (template context) - это объект или набор данных, которые передаются в шаблон для заполнения динамических элементов. Контекст шаблона предоставляет значения для переменных, используемых в шаблоне, и определяет, как данные будут отображаться при рендеринге.
4. Макет (layout) - это шаблон, который определяет общую структуру и разметку для нескольких страниц или представлений веб-приложения. Макеты содержат общие элементы, такие как шапка, подвал, боковая панель и другие компоненты, которые используются на разных страницах. Они позволяют упростить разработку и обновление внешнего вида приложения, поддерживая единообразный стиль и компоновку.
5. Частичное представление (partial view) - это фрагмент шаблона, который может быть включен или использован в другом шаблоне. Частичные представления позволяют избежать дублирования кода и повторного использования одних и тех же элементов в разных местах. Они могут быть использованы для отображения повторяющихся компонентов, таких как навигационное меню, боковая панель или комментарии.
6. Хелпер (helper) - это функция или класс, которая предоставляет дополнительную функциональность в шаблонизаторах или фреймворках. Хелперы используются для выполнения специфических операций или преобразований данных во время рендеринга шаблонов. Они могут использоваться для форматирования данных, выполнения математических операций, обработки строк и других задач.
7. Встроенные хелперы Handlebars - Handlebars предоставляет несколько встроенных хелперов, включая if, unless, each, with, lookup, log и другие. Эти хелперы позволяют выполнять условную логику, итерацию по данным, доступ к значениям и отладку внутри шаблонов.
8. Статический файл - это файл, который сохраняется на сервере и возвращается клиенту без изменений при каждом запросе. Статические файлы могут включать изображения, CSS-файлы, JavaScript-файлы и другие ресурсы, которые не требуют динамической обработки на сервере.
9. HTTP-заголовки:

* Cache-control - указывает, как кешировать и хранить промежуточные копии ресового файла на стороне клиента или прокси-сервера.
* ETag - идентификатор, представляющий конкретную версию ресурса. Используется для проверки целостности ресурса при кешировании.
* Last-Modified - дата и время последней модификации ресурса. Используется для проверки обновлений ресурса.
* If-Modified-Since - заголовок, содержащий дату последнего изменения ресурса. Используется для проверки, был ли ресурс изменен после указанной даты.
* If-None-Match - заголовок, содержащий ETag ресурса. Используется для проверки, соответствует ли текущая версия ресурса указанному ETag.
* Expires - указывает дату и время, после которых ресурс считается устаревшим и должен быть перезапрошен с сервера.

1. Свойство \_\_dirname объекта global - это строка, представляющая абсолютный путь к текущему рабочему каталогу файла JavaScript. Это свойство обычно используется в Node.js для определения пути к файлам и каталогам относительно текущего файла.
2. Метод response.download() - это метод объекта ответа (response) в веб-разработке, который отправляет файл для загрузки на клиентскую сторону. Он принимает путь к файлу и опциональное имя файла, которое будет отображаться на клиенте при сохранении файла. Метод устанавливает соответствующий заголовок Content-Disposition, чтобы указать браузеру, что файл должен быть загружен, а не открыт в браузере.
3. Метод response.attachment() - это метод объекта ответа (response), который устанавливает заголовок Content-Disposition с параметром "attachment". Это указывает браузеру, что ответ должен быть скачан как файл, а не показан в браузере. Метод не отправляет файл на клиентскую сторону, он только устанавливает соответствующий заголовок.
4. HTTP-заголовок Content-Disposition: attachment указывает браузеру, что сервер отправляет файл для загрузки, а не для просмотра в браузере. Браузер должен предложить пользователю сохранить файл на диск, обычно с использованием указанного имени файла в заголовке. Это позволяет создавать скачиваемые файлы, такие как документы, изображения или медиафайлы, и предлагать их пользователю для сохранения на локальном устройстве.