### Контрольные вопросы

1. Какими способами можно отлаживать адаптивную верстку сайта?
2. Какие основные инструменты для отладки клиентского JavaScript кода предоставляет браузер?
3. Как веб-приложение может хранить информацию на стороне клиента?

1. Для отладки адаптивной верстки сайта можно использовать следующие способы:

- Инструменты разработчика в браузере: современные браузеры предоставляют мощные инструменты разработчика, позволяющие анализировать и отлаживать адаптивную верстку. С их помощью можно изменять размер окна браузера, эмулировать разные устройства и проверять, как сайт реагирует на разные разрешения экрана.

- Медиа-запросы: в CSS вы можете использовать медиа-запросы для определения стилей, зависящих от разрешения экрана. С их помощью возможно добавлять и отлаживать медиа-запросы, чтобы убедиться, что верстка адаптируется корректно.

- Проверка на реальных устройствах: для полной уверенности в адаптивности сайта, также можно проверить его на реальных мобильных устройствах и планшетах.

2. Основные инструменты для отладки клиентского JavaScript кода, предоставляемые браузером:

- Инструменты разработчика: по большей части, инструменты разработчика браузера (например, Chrome DevTools) предоставляют мощный отладчик JavaScript. Есть возможность использовать точки остановки, просматривать значения переменных, отслеживать выполнение кода по шагам и многое другое.

- Консоль: в браузере есть JavaScript-консоль, где можно выводить отладочные сообщения и ошибки. Это помогает следить за выполнением кода и проверять значения переменных.

- Сетевой монитор: инструменты разработчика также предоставляют информацию о сетевых запросах, что позволяет отслеживать AJAX-запросы и убедиться, что они выполняются корректно.

- Profiler: некоторые инструменты разработчика предоставляют профилировщики производительности, которые помогают выявить проблемы с производительностью в вашем коде.

3. Веб-приложение может хранить информацию на стороне клиента с использованием следующих методов:

- Cookies — это небольшие текстовые файлы, которые могут храниться на стороне клиента и передаваться между сервером и браузером. Они часто используются для хранения маленьких данных, таких как настройки или идентификаторы сеансов.

- Web Storage включает в себя Local Storage и Session Storage. Эти API позволяют хранить данные на стороне клиента, но Local Storage сохраняет данные на долгосрочное время, в то время как Session Storage хранит данные только в рамках одной сессии браузера.

- IndexedDB — это более мощная база данных на стороне клиента, которая позволяет хранить структурированные данные. Она обеспечивает более сложные возможности хранения и запросов данных.

- Cache API позволяет веб-приложению кэшировать ресурсы, такие как изображения и стили, чтобы они могли быть доступны в автономном режиме.

- Web SQL Database: этот устаревший метод позволял использовать SQL-подобную базу данных на стороне клиента, но не рекомендуется к использованию.

Выбор метода хранения данных на стороне клиента зависит от конкретных потребностей веб-приложения и требований к безопасности данных.