

ОСНОВНЫЕ ИНСТРУМЕНТЫ ДЛЯ ТЕСТИРОВАНИЯ

Chrome (DevTools)

Firebug

Chrome DevTools — это набор инструментов, встроенных в браузер Google Chrome, для создания и отладки сайтов. С их помощью можно просматривать исходный код сайта, отлаживать работу frontend: HTML, CSS и JavaScript. Также DevTools позволяет проверять сетевой трафик, быстродействие сайта и многое другое.

Открыть инструменты для разработчиков несколькими способами:

Нажмите на иконку меню справа от панели навигации, кликните по пункту «Веб-разработчика», «Инструменты разработчика».

Используйте комбинации клавиш Shift + F5 (Windows и macOS) для запуска инструмента «Производительность» и Ctrl + Shift + E (Windows) или Cmd + Option + E (macOS) для запуска инструмента «Сеть».

Кликните правой кнопкой мыши по любому месту страницы и выберите в контекстном меню пункт «Исследовать элемент».

Пункты DevTools и их важность.

1. **Эмуляция android и ios, подключение android при отладке.** Есть необходимость проверять свои сайты на мобильных устройствах без их физического присутствия. Минус данного подхода заключается в том, что большинство этих эмуляторов являются коммерческими. Второй подпункт позволяет без всяких эмуляторов отследить все запросы и поведение Вашего продукта на устройствах android, просто подключив его к компьютеру и произведя небольшое количество манипуляций. Также плюс этого способа заключается в том, что можно настроить доступ к локальным серверам через такой тип подключения.
2. **Переопределение геолокации и подмена User-Agent.** В вышеуказанных двух пунктах говорится о возможности изменять (подменять) геолокацию нахождения устройства и параметры юзер агента. Подмена User-Agent поможет воспроизвести тот или иной баг, который был воспроизведен из какой-либо версии браузера или ОС. Закончив тестирование, никогда не забывайте возвращать данные User-Agent в исходное положение.
3. **Определение JS пути к строке.** Автоматизацией тестирования. Безусловно, данный способ не самый популярный для автоматизаторов, потому что этот путь может часто меняться, но на первых порах, когда еще не будет выработан скилл, помогающий с закрытыми глазами строить нужные селекторы для тестов, то эта возможность в DevTools может Вам помочь.
4. **Изменение стилей CSS у элементов.** В данном пункте рассматривается добавление фонового окраса для поля. Помимо этого, для элементов можно изменять и другие параметры (шрифт, размер, цвет и т.д.), для того чтобы можно было сразу указать разработчику или дизайнеру, как Вы видите этот элемент в контексте страницы либо, по просьбе заказчика изменить кнопку в “live” режиме.

5. [Неиспользуемые CSS и Javascript в вёрстке](#). Тестирование производительности, данный пункт будет интересен именно с точки зрения ускорения загрузки Вашего web-сайта. Если количество неиспользуемого кода, который каждый раз “пробегают” при загрузке той или иной страницы, очень велико, то при помощи действий, описанных в статье, будет возможность найти весь неиспользуемый код и указать его, как артефакт в баг репорте.
6. [Bebug JavaScript](#). Код - это вещь, которой в основном занимаются разработчики для отладки кода. Безусловно для отладки кода необходимо уметь его читать. Тестирование сталкивается с дебагом кода именно в Chrome.
7. [Сохранение изменений в Chrome](#) при перезагрузке страницы. Такую возможность добавили в DevTools относительно недавно (с 65 версией). Она позволяет сохранять все изменения, которые были внесены в те же CSS стили.
8. [Имитация медленного сетевого соединения](#). Эффект DevTools, который демонстрирует, как ведет себя страница при её загрузке на мобильных устройствах. Разработка, в 90% случаев, ведется при хороших условиях, связанных со скоростью интернета и его стабильностью, также как и тестирование, в связи с чем выловить баг, который воспроизводится у пользователя, становится возможным только полностью воспроизведя все окружение, в котором этот баг проявился.
9. [Настройка столбцов на вкладке Network](#). Можно настроить именно те столбцы, которые необходимы для анализа запросов на сайте в Вашем конкретном случае.
10. [Снимки экрана](#) при загрузке страницы.
11. [Блокирование запросов](#).
12. [Все о cookies](#) во вкладке applications. Cookies - очень важная вещь для анализа пользовательских сессий, соответственно, так как мы (тестировщики) воспроизводим пользовательские сценарии, нам необходимо знать, как можно улучшить тест-кейсы, используя работу с куками. В статье описаны все сценарии (поиск, удаление, изменение данных).

Firebug – полезный инструмент для web-разработчика (устарел, не существует на 2022 год)

Вызов Firebug происходит несколькими способами:

через меню Инструменты > Firebug;

через горячую клавишу **F12**; щелчком по картинке жука в правом углу браузера Firefox, картинка доступна только при включенной строке состояния (Вид > Строка состояния).

Также плагин **firebug** предоставляет следующий функционал:

Инспектор (**Inspect**)— функция, предназначенная для быстрого обнаружения тега и просмотра всех «*привязанных*» к нему свойств и стилей;
Возможность редактировать **HTML**-код и таблицы стилей **CSS** непосредственно в окне браузера. Можно вносить различные изменения в атрибуты тега и сразу же наблюдать результат. Это очень удобно для нахождения нужного оформления;
Возможность отладки **JavaScript**;
Дополнительный функционал, который позволяет просмотреть процесс загрузки страницы.

Это далеко не полный список функций, которые представляет плагин для web-разработчиков. Помимо всего прочего были созданы дополнения к **firebug**:

Firecookie — инструмент для просмотра и управления **cookies**;

FirePHP — инструмент для отладки **PHP**-кода при помощи **HTTP**-заголовков;

FireRainbow — инструмент, который выделяет (*подсвечивает*) вставки **JavaScript**;

Page speed firebug — дополнение, позволяющее измерить и оптимизировать скорость загрузки сайта..

Исходный код

В отличие от обычного исходного кода HTML, Firebug учитывает изменение кода после его преобразования через JavaScript. Таким образом, вы сможете отследить элементы, добавляемые через скрипты, и посмотреть стиль, которые невозможно обнаружить иначе.

Быстрое редактирование HTML

С помощью Firebug можно вносить изменения непосредственно в код документа и видеть результат их воздействия немедленно. Вы можете удалять, добавлять, редактировать атрибуты тегов, просто щелкая по ним.

Инспектирование кода CSS

Для любого выбранного элемента показывается его стиль, отдельные свойства при этом можно отключать, добавлять новые свойства, менять значения существующих. Результаты изменений отображаются в браузере немедленно.

Поиск любого элемента мышкой

С помощью инструмента «Анализировать элемент» вы можете выбрать любой текст, изображение или объект на странице, просто щелкнув по нему. При этом откроется HTML-код выбранного элемента и его стиль.

Стиль

Firebug открывается в нижней части браузера и его окно состоит из двух основных панелей и меню. В левой панели по умолчанию отображается HTML, а в правой стиль выбранного элемента.

Код HTML выводится в виде иерархической структуры, если элемент содержит дочерние элементы, рядом с ним выводится плюсики. Любой элемент можно выбрать не только через код HTML, но и непосредственно на странице с помощью инструмента **Анализировать элемент** либо через меню **Анализировать элемент** (**Ctrl + Shift + C**).

Редактировать стиль элемента

Для элемента установить встроенный стиль (он включается через атрибут **style**).

Новое свойство

Добавить новое стилевое свойство и его значение.

Редактировать

Изменить выбранное свойство.

Удалить

Удалить выбранное свойство

Выключить

Отключить стилевое правило

Обновить

Обновить стиль. Даёт результат при модификации стилевого файла внешней программой.

Анализировать во вкладке CSS

Перейти во вкладку CSS.

Анализировать во вкладке DOM

Перейти во вкладку DOM. Отключение свойства, изменение их значений и любые другие манипуляции со стилями сразу же отображаются на веб-странице. Если страницу обновить, все действия в Firebug отменяются. Благодаря этому можно экспериментировать с разными свойствами, не опасаясь принести вред документу.

Скомпилированный стиль

Для каждого элемента документа браузер автоматически формирует свой стиль, основываясь на внутренних значениях браузера и стиле родительских элементов. Во вкладке «Скомпилированный стиль» приведены свойства и их значения, которые явно не задаются через стили. Скомпилированный стиль нельзя изменить, и он приведён лишь для информации.

Макет

В этой вкладке для выбранного элемента показывается его блочная модель, включающая ширину и высоту элемента, значение полей, границ, отступов и позиционирования.