

7. Council Directive 89/391/EEC of 12 June 1989 on the introduction of measures to encourage improvements in the safety and health of workers at work // Official Journal of the European Communities. – 1989. – L 183. – P. 1–8.

8. Ethics guidelines for trustworthy AI // European Commission, High-Level Expert Group on Artificial Intelligence. – Brussels, 2019. – 41 p.

9. Arbeitsschutzgesetz (ArbSchG) vom 7. August 1996 (BGBl. I S. 1246), das zuletzt durch Artikel 12 des Gesetzes vom 10. August 2021 (BGBl. I S. 3436) geändert worden ist.

10. Workplace Safety and Health (WSH) 2028 Strategy // Ministry of Manpower Singapore. – Singapore, 2022. – URL: <https://www.mom.gov.sg> (дата обращения: 15.04.2024).

11. Об информации, информатизации и защите информации : Закон Респ. Беларусь от 10 ноября 2008 г. № 455-3 (в ред. от 01.01.2024) // Нац. реестр правовых актов Респ. Беларусь. – 2024. – № 2/3040.

УДК 004.42

CSS HOUDINI: АНАЛИЗ ПРАКТИЧЕСКОГО ПРИМЕНЕНИЯ И БАРЬЕРОВ ВНЕДРЕНИЯ В СОВРЕМЕННОЙ ВЕБ-РАЗРАБОТКЕ

Кудрявцева А. В., Шутько Н. П.

студент; к.т.н., доц., БГТУ

Введение. История веб-стандартов показывает, что внедрение новых CSS-функций, вроде Flexbox, занимает много времени. Старые JavaScript-полифилы, работающие после рендеринга и вызывающие пересчет стилей, часто замедляют работу сайта.

Основная часть. CSS Houdini меняет подход, предлагая набор API низкого уровня для браузеров. Он позволяет JavaScript-разработчикам расширять CSS, используя лёгкие Worklets для работы с ключевыми этапами рендеринга: Style, Layout, Paint и Composite. Это открывает доступ к API, таким как Paint API (для создания фонов и рамок программно), Layout API (для собственных режимов раскладки), Animation Worklet (для быстрых анимаций) и Properties & Values API (для типизации CSS-переменных) [1]. Houdini даёт возможность создавать быстрые полифилы и пробовать будущие стандарты прямо в процессе рендеринга, а не отдельно, что решает проблему производительности.

Несмотря на изначальный потенциал, Houdini, спустя годы после появления первых спецификаций, всё ещё ограниченно применяется в коммерческих продуктах. В докладе анализируются существующие, а не гипотетические, варианты применения

CSS Houdini и рассматриваются причины, тормозящие его распространение.

В настоящее время Houdini ещё не получил широкого распространения в обычной разработке интерфейсов. Он полезен там, где играют ключевую роль точный контроль и производительность, но сложен в применении. Эта технология пока экспериментальная и чаще применяется для создания отдельных, в основном демонстрационных, эффектов, а не для построения всего интерфейса целиком.

Прежде всего, он подходит для создания необычных визуальных эффектов. Его часто используют для сайтов-визиток, портфолио и демонстрационных проектов, где нужен яркий и запоминающийся эффект. Paint API хорош для создания динамических фонов, сложных узоров и интерактивных элементов, которые сложно сделать только на CSS. Например, можно сделать фон, который меняется в зависимости от положения мыши или времени суток. В этом случае Houdini больше подходит для дизайнеров, чем для обычных разработчиков интерфейсов.

Кроме того, Houdini помогает создавать прототипы и дополнения для будущих веб-стандартов. Идея в том, чтобы разработчики могли пробовать новые функции до того, как они станут общепринятыми. Хороший пример – использование Properties & Values API для анимации CSS-переменных. Обычно CSS-переменные нельзя анимировать стандартными средствами, но с Houdini можно зарегистрировать переменную как `<color>` и сделать так, чтобы градиент или тень плавно менялись. Так Houdini помогает расширить возможности CSS. Однако более сложные дополнения, например, для создания полноценной masonry-раскладки через Layout API, пока остаются скорее идеями из-за недостаточной поддержки.

Наконец, Houdini может помочь повысить скорость работы в некоторых задачах. Например, если обычные JS-анимации работают медленно (параллакс-эффекты при прокрутке), можно использовать Animation Worklet. Он работает в отдельном процессе, поэтому анимации воспроизводятся без задержек, даже если на текущей процедуре большая нагрузка. Но это полезно только для проектов, где очень важна плавная анимация.

Несмотря на преимущества, есть объективные причины, по которым технология не вышла из стадии смелых экспериментов.

Во-первых, проблема с внедрением технологии в том, что браузеры поддерживают её по-разному. К примеру, в 2020 году основные

API, такие как Paint и Properties & Values, работали нормально только в браузерах Blink [2]. К концу 2024 – началу 2025 года полная поддержка появилась в основном также в Blink (Chrome, Edge, Opera) и WebKit (Safari), а в Firefox все еще наблюдаются недостатки или проводится тестирование [3]. Это значит, что для работы с технологией в коммерческих проектах, где важна поддержка разных браузеров, приходится создавать альтернативные механизмы со сложной реализацией, что часто делает технологию не вполне практичной. Поэтому, прежде чем использовать Houdini в реальных проектах, важно проанализировать, как конкретные API поддерживаются разными браузерами, например, на сайте Can I use [4].

Во-вторых, для работы с Houdini нужно хорошо понимать не только JavaScript, но и как браузер обрабатывает рендеринг. Создание Worklet'ов, особенно для Layout API, намного сложнее, чем написание обычного CSS или даже JS-анимации. Отладка кода Worklet также представляет трудности, и неправильный код может ухудшить производительность.

В-третьих, острой потребности в этом нет. Для большинства задач (сайты компаний, онлайн-магазины, приложения) достаточно возможностей CSS и хорошей оптимизации. Затраты на обучение команды, внедрение и поддержку этой технологии обычно не оправдывают себя с финансовой точки зрения.

Заключение. CSS Houdini – это технология, кардинально меняющая подход к рендерингу, несмотря на то, что пока её использование ограничено из-за неполной поддержки в разных браузерах и сложности работы с ней. Широкое внедрение станет возможным только после унификации поддержки и появления удобных инструментов, скрывающих её сложность. Пока же Houdini остаётся мощным, но специфическим инструментом для экспериментов.

Список использованных источников

1. Habr [Электронный ресурс] – Режим доступа: <https://habr.com/ru/articles/494660/> – Дата доступа: 07.12.2025.
2. Github [Электронный ресурс] – Режим доступа: <https://github.com/CSSHoudini/awesome-css-houdini> – Дата доступа: 07.12.2025.
3. Tenchat [Электронный ресурс] – Режим доступа: <https://tenchat.ru/media//4054687-css-houdini-2025-programmirovem-stili-na-chistom-javascript> – Дата доступа: 07.12.2025.
4. Can I use [Электронный ресурс] – Режим доступа: <https://caniuse.com/> – Дата доступа: 07.12.2025.