


КАК СТАТЬ АВТОРОМ

Неделя тестировщиков

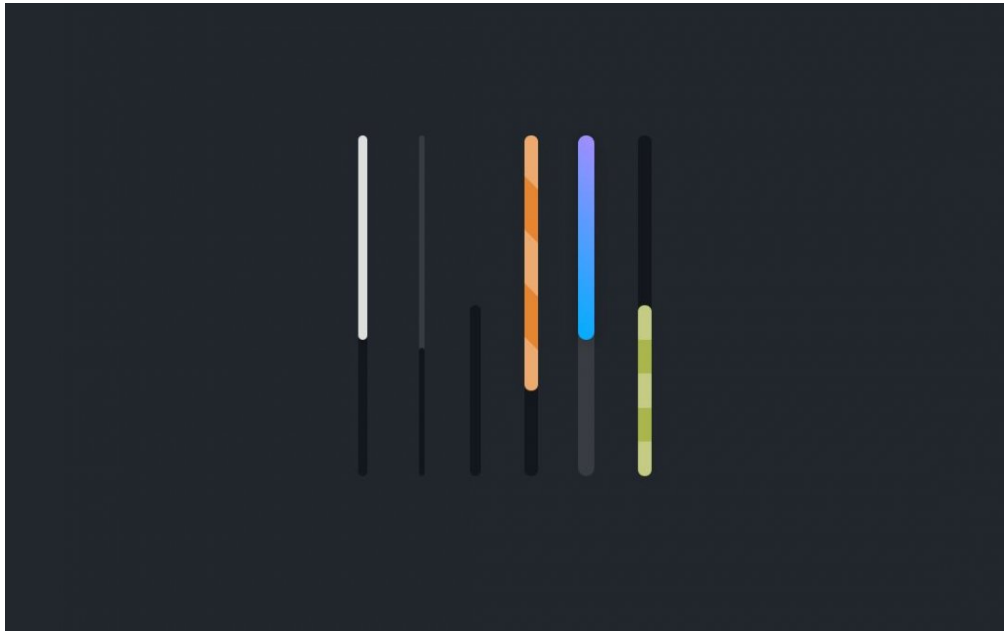
Знаешь Java? Держи квест!

 abaha91 30 июня в 20:37

## Scrollbar в современном CSS

Разработка веб-сайтов\*, CSS\*, JavaScript\*, Разработка мобильных приложений\*

Перевод

Автор оригинала: <https://hashnode.com/@mayxnk>

### Scrollbar это больше, чем принято считать

Всем привет! Начнем с того что scrollbar это очень обыденный элемент, присутствующий на всех веб-сайтах и в каждом приложении. Исторически сложилось так, что его было трудно кастомизировать и поэтому разработчики старались обходить стороной работу с ним если того четко не требовало ТЗ, но все же, как мне кажется, мы должны уделять им немного больше внимания. Давайте посмотрим, что мы можем сделать в 2022 году, чтобы улучшить возможности прокрутки для наших пользователей.

#### Основы

Когда содержимое элемента слишком велико и не помещается в нем, то, чтобы сделать его прокручиваемым, мы можем использовать `overflow: auto`. Это нужно, чтобы полосы прокрутки отображались по умолчанию.

Что касается стилизации scrollbar-a, то вы можете использовать свойства `scrollbar` с префиксом `-webkit` в сочетании со стандартными свойствами ширины: `scrollbar-width` и цвета: `scrollbar-color`. На эту тему есть множество публикаций, поэтому я не буду вдаваться в подробности. Если вы еще не знакомы с этими свойствами, я рекомендую вам ознакомиться с [руководством](#), составленным Ахмадом Шейдидом.

Так же я хочу упомянуть плагин `postcss-scrollbar`, который генерирует свойства полосы прокрутки с префиксом `-webkit` из стандартных свойств, что позволяет вам получить кросс-браузерные стили только с помощью допустим такого кода:

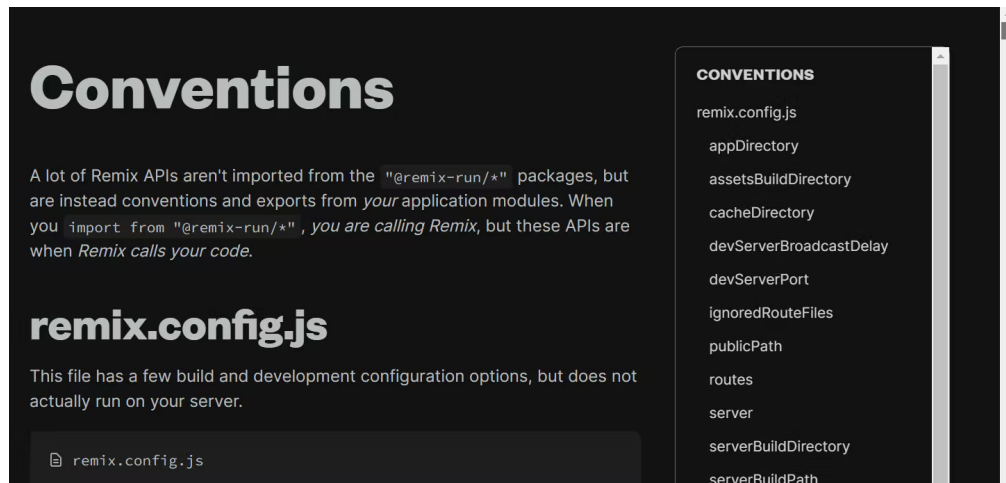
```
.scroll-container {
  overflow: auto;
  scrollbar-width: thin;
  scrollbar-color: hsl(0 0% 50%);
  /* postcss-scrollbar will add the -webkit version automatically! */
```

```
}
```

Выглядит достаточно просто. А теперь перейдем к более интересным моментам.

### Scrollbar и темы

Если вы все еще хотите использовать стили scrollbar по умолчанию, это ваше право. Но вам нужно учитывать, если ваш сайт поддерживает темную тему. На многих сайтах при переключении на темный режим полосы прокрутки застревают в светлом. Вот как выглядят документы remix.run в Windows в темном режиме:



Это можно исправить с помощью свойства `color-scheme`. Прибегать к ним вам придется в разных ситуациях, в том числе – не связанных со scrollbar. У Томаса Штайнера есть [отличная статья](#) на web.dev, если вы хотите узнать об этом больше.

```
html {
  /* defer to OS preference */
  color-scheme: dark light;
  /* override, assuming the theme toggler sets a data-theme attribute */
  &[data-theme=light] { color-scheme: light; }
  &[data-theme=dark] { color-scheme: dark; }
}
```

### Особенности полосы прокрутки в различных операционных системах

В данном разделе я буду размышлять о различных окружениях, поведении в них полос прокрутки и проблемах, связанных с тем или иным поведением, а также выскажу некоторые свои предположения. Если вас интересует только код, смело прокручивайте этот раздел до конца.

#### Windows

Как я продемонстрировал в предыдущем разделе, Windows всегда отличалась самыми уродливыми полосами прокрутки, по сравнению с другими платформами. Windows 10 имеет наиболее неприглядные полосы прокрутки во всех браузерах. Даже если выбрать правильную цветовую схему, они все равно выглядят излишне квадратными и толстыми.

В Windows 11 попытались модернизировать полосы прокрутки, сделав их тоньше и накладывая друг на друга. Что касается браузеров, Firefox загружает их в таком виде всегда, в то время как Edge или Chrome – модифицируют в той или иной форме.

Особенно мне не нравится реализация Firefox, потому что полоса прокрутки тонкая и исчезает, если вы активно не взаимодействуете с ней. Что еще хуже, кажется, что нет никакого способа изменить это поведение программным путем. Firefox поддерживает опцию «Всегда показывать полосы прокрутки» только в Windows 11, но она отключена по умолчанию, и естественно большинство пользователей не догадываются ее включить. Я очень надеюсь, что этот момент будет

учтен в следующем обновлении браузера. Но пока мы ничего сделать с этим не в состоянии.

В Chromium все проще, там у нас гораздо больше возможностей повлиять на scrollbar, хотя бы с теми же стилями `-webkit-scrollbar`.

## macOS

Полосы прокрутки в macOS выглядят менее безобразными, но они имеют свои недостатки.

Возможно, я слишком придирчив, но я разочарован, что не могу просто переместить курсор на правый край окна браузера и использовать его для прокрутки. Это связано с тем, что macOS резервирует края для изменения размера или перемещения окна, даже если курсор все еще находится на бегунке полосы прокрутки. Это неизбежно делает и без того тонкую полосу прокрутки еще тоньше (хотя и только для прокрутки окна просмотра).

Вторая проблема более актуальна: автоматически исчезающие полосы прокрутки. Они могут способствовать созданию приятного и не загроможденного интерфейса, но... это в том случае если пользователь знает, что часть страницы является прокручиваемой. Недавно у меня был случай, когда я буквально не мог найти билет на конференцию, потому что он был скрыт за пределами обозримой области, а полоса прокрутки была невидимой. Другими словами, исчезновение полос прокрутки может в буквальном смысле стоить вам денег. Чтобы обойти эту проблему, нужно высоту полосы прокрутки сделать такой, чтобы последний элемент был наполовину виден. Или же – использовать JavaScript для определения положения прокрутки и добавления тени прокрутки.

Но ест и хорошие новости, обе эти проблемы можно избежать путем задания пользовательского стиля.

## Android, iOS и сенсорные экраны

У сенсорных устройств такие же тонкие, накладные и исчезающие полосы прокрутки, которые я назвал проблемой в macOS, но используются они совершенно по-другому: пользователь не взаимодействует напрямую с полосой прокрутки, а просто проводит пальцем по экрану. Кроме того, поскольку сенсорный экран обычно более маленький, на нем умещается меньше информации, и ожидается, что пользователь будет прокручивать намного чаще, чтобы просмотреть контент. Именно потому существуют специальные функции, такие как (1) прокрутка на основе импульса (инерция) и (2) визуальная подсветка или пружинная анимация при достижении края прокручиваемого контейнера. Android еще более внимательно относится к этому и показывает полосу прокрутки при первом появлении прокручиваемого элемента, а затем переключается на обычное автоматическое исчезновение полос после того, как пользователь прокрутил хотя бы один.

Мне очень нравится поведение по умолчанию этих scrollbar-ов и я хотел бы сохранить его. Это достижимо с помощью медиа-запроса `pointer`, который позволит нам условно стилизовать полосы прокрутки в зависимости от того, использует ли пользователь мышь или сенсорный экран.

```
@media (pointer: fine) {  
  .scroll-container {  
    /* ...custom scrollbar styles only for desktop */  
  }  
}
```

Я много где это использовал и мне честно говорят результат очень понравился.

## Предотвращение поехавшей верстки

Одна из проблем с `overflow: auto` заключается в том, что оно показывает полосу прокрутки только в том случае, если содержимое переполнено. Это означает, что внезапное появление полосы прокрутки может вызвать небольшое смещение верстки. Эту проблему можно убрать с помощью `overflow: overlay` (в Chrome), чтобы полоса прокрутки никогда не занимала место, и / или с помощью `overflow: scroll`, чтобы она всегда была на месте.

В настоящее время мы можем указать браузеру, чтобы он зарезервировал место для полос

прокрутки, используя `scrollbar-gutter: stable`. Он поддерживается в Chrome и Firefox. Если вы используете невидимый scrollbar, то этот резервный вариант должен работать так же, как `scrollbar-gutter`.

```
.scroll-container {
  overflow: scroll;
  @supports (scrollbar-gutter: stable) {
    overflow: auto;
    scrollbar-gutter: stable;
  }
}
```

Здесь стоит отметить, что если вы хотите, чтобы `scrollbar-gutter` находилась в области просмотра, то работать с `<body>` ее заставить сложно, и, возможно, будет проще переместить ее в дочерний элемент.

### Еще больше возможностей для scrollbar в современном CSS

Хотя это не связано напрямую со стилем полосы прокрутки, я хочу кратко упомянуть еще несколько свойств, которые могут помочь улучшить прокрутку.

#### scroll-padding

`scroll-padding` позволяет создать смещение по краю области прокрутки. Это полезно, когда используется фиксированный заголовок (sticky header), способный закрывать заголовки фрагментов, которые прокручиваются на странице с помощью ссылок перехода или URL-адресов.

```
.scroll-container {
  scrollbar-padding-top: var(--header-height);
}
```

#### scroll-behavior

`scroll-behavior` позволяет включить плавную прокрутку, которая также очень удобна для переходов по ссылкам-якорям внутри страницы.

```
@media (prefers-reduced-motion: no-preference) {
  .scroll-container {
    scrollbar-behavior: smooth;
  }
}
```

#### scroll-padding

`Overscroll-behavior` выполняет совершенно иную функцию: позволяет предотвратить цепочку прокрутки во вложенных областях прокрутки. Удобно в некоторых местах, таких как диалоговые окна и боковые панели, где вы не хотите, чтобы базовая страница начинала прокручиваться, когда вы достигаете конца текущего контейнера прокрутки.

```
.scroll-container {
  overscroll-behavior: contain;
}
```

#### Scroll snapping

Представленная некоторое время назад в CSS scroll snapping претерпела многочисленные улучшения и исправления. Сегодня это достаточно надежный и оптимальный способ создания чего-то вроде галереи или компонента stories с почти нулевым использованием JavaScript.

```
.scroll-container {  
  scroll-snap-type: x mandatory;  
  & > * {  
    scroll-snap-align: start;  
  }  
}
```

## Заключение

Благодарю за прочтение публикации и надеюсь, что она вдохновит вас идти по пути улучшения scrollbar на ваших проектах. А на этом собственно все.

Если вдруг кому интересно то я веду [телеграм канал по фронтенду](#) где выкладываю интересные статьи на разные темы а так же сам периодически пишу шорт-риды которые могут быть вам полезны.

**Теги:** [javascript](#), [scrollbar](#), [css](#)

**Хабы:** [Разработка веб-сайтов](#), [CSS](#), [JavaScript](#), [Разработка мобильных приложений](#)

👍 +6

👁 3.1K

📖 50



## Редакторский дайджест

Присылаем лучшие статьи раз в месяц



8

5.2

Карма

Рейтинг

**Геннадий @abaha91**

Frontend

[Telegram](#)

Реклама

## Комментарии 8



**php7**

30.06.2022 в 22:25

Расстрелял бы за однопиксельную полосу прокрутки.

👍 +7

[Ответить](#)



dartraiden  
01.07.2022 в 20:25

Чем это плохо? Зачем мне толстая полоса, если я не собираюсь туда попадать мышью?

— когда мне требуется попасть по ней мышью, полоса прокрутки становится толстой и позволяет легко это сделать

— когда контент прокручивается колесом мыши или клавиатурой, от полосы прокрутки мне требуется лишь индикация прогресса, и тонкая полоса с этим справляется

0 Ответить

funca  
30.06.2022 в 23:30

Есть два альтернативных подхода к пониманию роли стандартных элементов управления в web.

Дизайнерский - все везде должно выглядеть однообразно, как на макете. Попутно исправляем косяки отображения в разных браузерах.

Системный - внешний вид и поведение элементов управления должны соответствовать стандартам графического интерфейса операционной системы. (условно, если MacOS отображает скроллы тонкими во всех окнах, значит так надо).

Оба имеют право на жизнь. Но как только в требованиях появляется A11y, все скатывается ко второму потому, что учесть все массу нюансов для разных девайсов ни какого бюджета не хватит)

0 Ответить

Gigatrop  
30.06.2022 в 23:57

А мне полосы в Windows нравятся больше всего. По ним можно попасть без прицеливания и без ожидания увеличения размера. Они состоят из пяти элементов с разной функцией, а не только из ползунка. И везде одинаковые. Если скролл меньше 5 миллиметров по ширине линейкой на экране, то это декорация, а не функциональный компонент. Потому что тогда нельзя переместить на них курсор и прокрутить без прицеливания и задержки для увеличения (лишнее прерывание намерения, как если бы все кнопки были сначала миллиметр в высоту). А самые плохие видел в убунту, где по умолчанию прокрутку не видно, и нужно догадаться и проскроллить мышкой, чтобы появился какой-то квадратик всегда одинакового размера, на который не всегда удобно попадать за пределами окна:

Hidden text

+10 Ответить

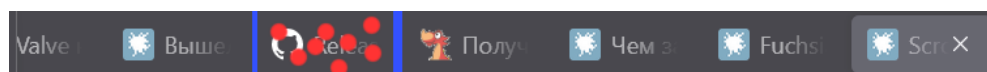
dartraiden  
01.07.2022 в 20:29

прокрутить без прицеливания и задержки для увеличения


Никакой задержки нет, оно появляется ровно в тот момент, когда мышь подходит в область, где должна быть полоса прокрутки.

Пользователь же не старая бабка, которая двумя руками, высунув кончик языка, старательно перемещает мышь и прицеливается. Это всё органично происходит, нафига куда-то целиться, мы в грёбаном тире или что?


Из той же оперы и пользователи, которые жалуются «я не вижу границы между вкладками и не могу в них попасть». Ядрён-батон, а что ты, дорогой пользователь, вообще там елозишь у краёв вкладок, как мексиканец у американской границы? Вон, там туева хуча места, куда можно кликнуть (красные точки), на кой ляд тебе эти приграничные зоны? (синие области)






-1 Ответить


-  **Gigatrop**  
01.07.2022 в 20:38





Где? У вас просто задача сказать наперекор?

 0 [Ответить](#)

 
-  **Gigatrop**  
01.07.2022 в 21:25


Откуда вы знаете за другого пользователя если не видите конкретного человека? Может и как старая бабка прицеливаются. Бывает что нужно на автомате что-то делать типа скролла, а глаза и мозг уже мутные под конец дня и любой обрыв автоматизма приводит к усилиям и негативной реакции. Представьте, бывают другие люди не такие как вы и не в таких идеальных состояниях и условиях.



 0 [Ответить](#)

 
-  **dartraiden**  
01.07.2022 в 20:20 

Особенно мне не нравится реализация Firefox, потому что полоса прокрутки тонкая и исчезает, если вы активно не взаимодействуете с ней.

Firefox лишь следует системным настройкам, и это правильно. Когда каждое приложение начинает своевольничать, у нас и получается классическое виндовое «кто в лес, кто по дрова» и никакой целостности в вопросах визуального оформления.

 0 [Ответить](#)


 

Только полноправные пользователи могут оставлять комментарии. [Войдите](#), пожалуйста.

#### ПОХОЖИЕ ПУБЛИКАЦИИ

28 января 2013 в 02:50

##### Пользовательский JavaScript и CSS на мобильных устройствах

 +18  17K  140  27 +27


7 января 2012 в 18:14

##### Параллельная загрузка JavaScript и CSS без блокирования парсинга страницы

 +84  65K  540  49 +49

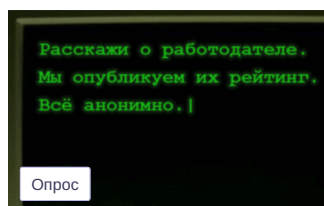
20 июня 2007 в 19:59

##### Объединение JavaScript и CSS в одном файле

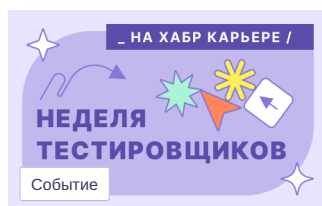
 +18  12K  30  93 +93

#### МИНУТОЧКУ ВНИМАНИЯ

[Разместить](#)



Третье хабраисследование ru-IT-брендов



Неделя тестировщиков на Хабр Карьере



Жизненный квест про работу в банке

#### ВАКАНСИИ

JavaScript разработчик

от 100 000 ₽ · Сбер · Иннополис

JavaScript разработчик

от 70 000 до 150 000 ₽ · НПО Фарватер · Москва · Можно удаленно

JavaScript Developer (Senior frontend)

от 250 000 до 400 000 ₽ · EVERA PEOPLE · Можно удаленно

Web-программист (Bitrix) / PHP, CSS, JS (Fullstack)

от 80 000 ₽ · VIARDA · Краснодар

Mid - Senior Frontend Developer (Javascript/VueJS/Remote/Relocation)

до 5 000 \$ · CoverGo · Можно удаленно

[Больше вакансий на Хабр Карьере](#)

#### ЛУЧШИЕ ПУБЛИКАЦИИ ЗА СУТКИ

вчера в 11:16

**Bash отладчик с поддержкой произвольных точек останова**

♦ +44

👁 4.1K

🔖 74

💬 5 +5

вчера в 14:40

**Новые нули дзета-функции**

♦ +31

👁 4K

🔖 14

💬 10 +10

вчера в 21:20

**Личный опыт выгорания**

♦ +26

👁 8K

🔖 31

💬 23 +23

вчера в 17:00

**DIY квантовые вычисления: как я начал собирать квантовые схемы**

♦ +21

👁 8.2K

🔖 41

💬 22 +22

вчера в 21:00

**Антиматерия и бариогенезис. Три причины, почему нет антивещества, но есть мы**

♦ +15

👁 2.8K

🔖 21

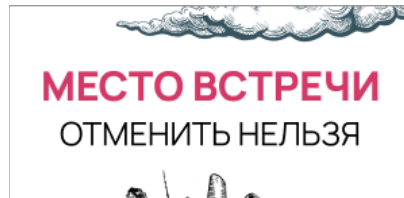
💬 1 +1

**Хотим узнать, что вы думаете о своём месте работы (анонимно)**

Опрос

Реклама





ЧИТАЮТ СЕЙЧАС

Эксперт пояснил, что обнаруженная «Персеверанс» сброшенная кожа ксеноморфа — это часть «Небесного крана»

39K 39 +39

Завершился международный изоляционный эксперимент SIRIUS-21 длиной 240 суток

12K 17 +17

Украинский изобретатель собрал рабочий «математический велосипед»

25K 44 +44

Личный опыт выгорания

8K 23 +23

Теория выученной беспомощности. Что это и откуда она взялась

3.9K 23 +23

Ну-ка повтори: почему разработчики и дизайнеры не понимают друг друга

Мегапост

РАБОТА

React разработчик  
151 вакансия

JavaScript разработчик  
346 вакансий

Swift разработчик  
86 вакансий

Android разработчик  
78 вакансий

iOS разработчик  
70 вакансий

Ваш аккаунт

Войти  
Регистрация

Разделы

Публикации  
Новости  
Хабы  
Компании  
Авторы  
Песочница

Информация

Устройство сайта  
Для авторов  
Для компаний  
Документы  
Соглашение  
Конфиденциальность

Услуги

Корпоративный блог  
Медийная реклама  
Нативные проекты  
Образовательные программы  
Мегaproекты



[Настройка языка](#)

[Техническая поддержка](#)

[Вернуться на старую версию](#)

© 2006–2022, Habr