



HTML и CSS. Адаптивная вёрстка и автоматизация

Уровень 2, с 21 ноября 2022 по 30 января 2023

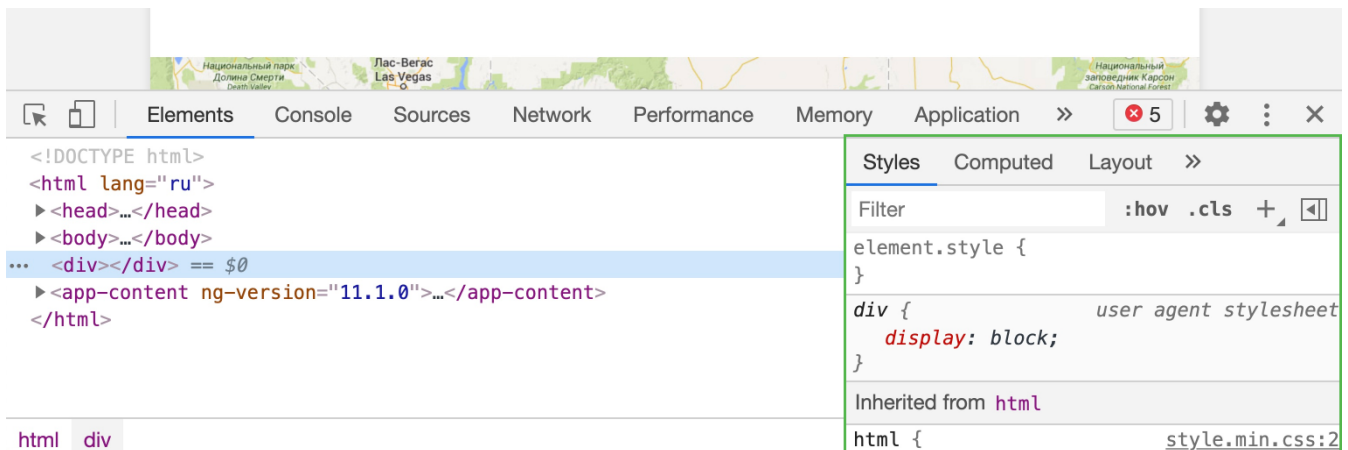
Меню курса

[Главная](#) / [3. Препроцессоры и автоматизация](#) /

📖 3.18. Вкладка Styles

🕒 ~ 8 минут

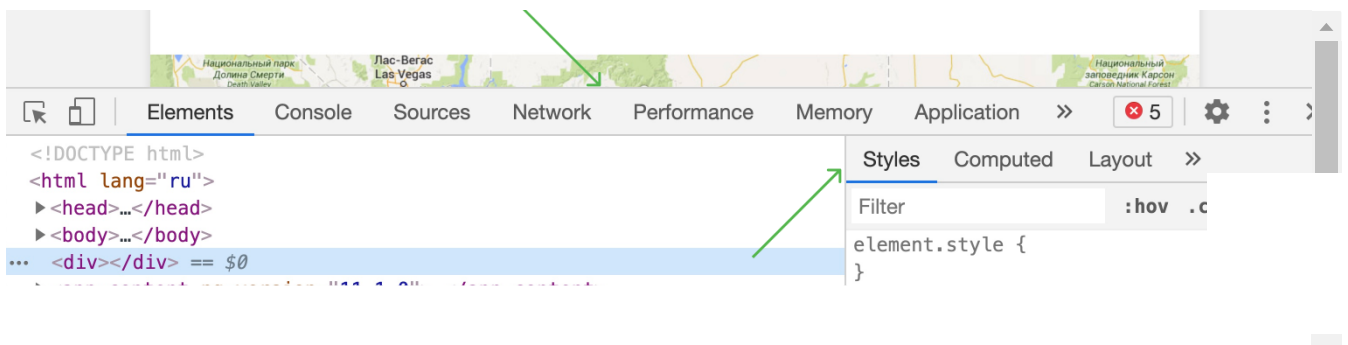
Во время работы с DOM-деревом часто требуется выяснить, какие стили установлены на тот или иной элемент. В этом помогает вкладка Styles:



Выбор вкладки Styles

Если во вкладке (или в целом в панели инструментов разработчика) не хватает места или не видно какие-либо участки, можно:

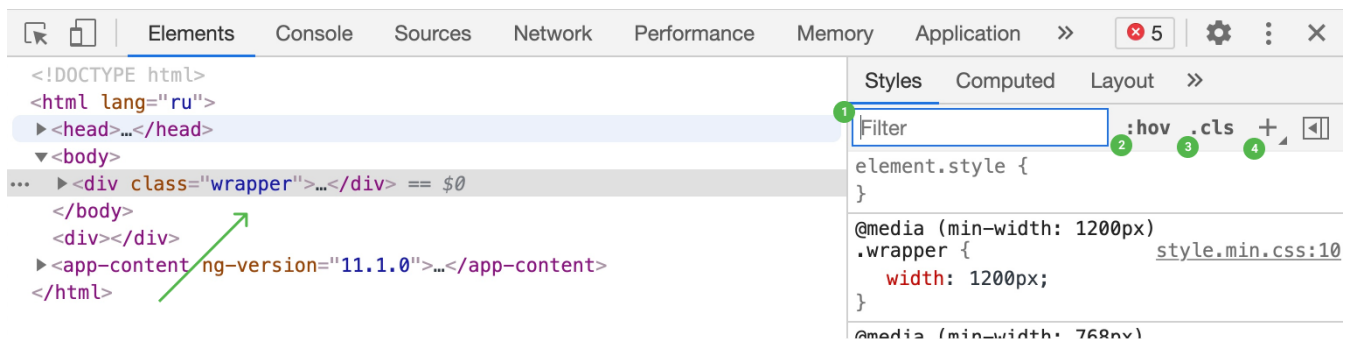
1. Увеличить размеры той или иной вкладки или всей панели инструментов, потянув мышкой за края:



Границы панели, доступные для изменения размеров

2. Увеличить шрифт и элементы в инструментах разработчика, нажав комбинацию **Ctrl +** на клавиатуре (уменьшить — **Ctrl -**, восстановить в исходное состояние — **Ctrl 0**). Если используете macOS — клавиша **Ctrl** заменяется на **Command**.

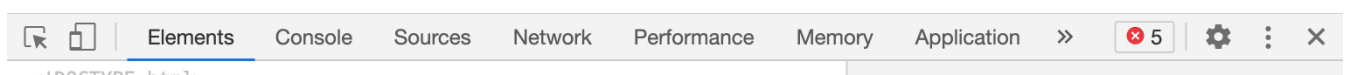
Перейдем к вкладке Styles и посмотрим на её возможности. Для начала необходимо выбрать любой элемент DOM-дерева:



Выбираем элемент

И теперь по порядку разберём каждый элемент верхнего меню:

1. Поиск по селектору. Здесь можно найти селектор, который применяется к выбранному элементу. Для начала просто прокрутим вниз и увидим, что к данному элементу применяется селектор **body**:



Список селекторов

Если ввести в поле поиска селектор **body**, то отобразится именно этот селектор и подсветится жёлтым цветом:

Поле для поиска селектора

2. `:hov` открывает выпадающее меню, в котором можно выбрать псевдокласс и применить его к элементу:

Список псевдоклассов

То же самое можно сделать через DOM-дерево, нажав на элементе правой кнопкой мыши и выбрав нужный пункт в меню `Force state`.

3. `.cls` открывает меню с возможностью ввести новый класс для элемента и применить или удалить существующий класс у элемента:

Управление классами

« Замечание: если галочка напротив класса была снята, то браузер запомнит, что на данном элементе был этот класс. Если галочку снять, поработать где-то в другом месте, а потом вернуться в пункт `.cls`, то можно будет вновь установить галочку на тот класс (или классы), что был у элемента изначально.

4. `+` добавит новый селектор в таблицу стилей. Можно вписать любой селектор:

Добавляем класс «wrapper»

Теперь перейдём к блоку с селекторами:

Блок с селекторами

Селекторы и путь к файлу стилей

В этом блоке перечислены селекторы и их правила. Если селекторы находятся в @media-выражениях, то будут показаны и они:

Пример @media-выражения

При нажатии на селектор левой кнопкой мыши можно перейти в режим редактирования селектора и изменить его.

Справа от селектора указан путь до файла и строка, в которой записан данный селектор. При нажатии на путь можно попасть в панель `sources` и редактировать ваш CSS в ней как обычный файл:

Встроенный редактор кода

Если вместо пути до файла написана строка `user agent stylesheet`, значит данные правила были применены браузером и их нельзя изменить из панели инструментов разработчика.

Свойства и значения

В этой панели можно увидеть, какие правила были переопределены. Например, правило `width: 100%;` зачёркнуто, так как оно было переопределено правилом `width: 1200px;` из-за того, что `width: 1200px;` находится ниже по коду, чем `width: 100%;`

Правило не действует

Галочка напротив правила отключает действие самого правила.

Сайт до отключения правила

Сайт после отключения правила

Нажатие на иконку `+` дублирует селектор.

Нажатие левой кнопкой мыши на значение правила переводит его в режим редактирования значения:

Значение можно редактировать

Режим редактирования значения поддерживает подсказки при вводе, как и любой редактор кода:

Выбор из подсказки

Выбор значения осуществляется стрелками вверх и вниз на клавиатуре. Чтобы применить выбранное значение, нужно нажать на стрелку вправо (этот способ дополнит правило, но не выйдет из режима редактирования) или клавишу Enter — этот способ дополнит правило и выйдет из режима редактирования или перейдёт к следующему свойству.

Нажатие левой кнопкой мыши на свойство действует так же, как и нажатие на значение правила, и здесь тоже поддерживаются подсказки.

Нажатие в пустое пространство приведёт к отображению курсора на месте ввода свойства, и можно начать вводить нужные свойства с необходимого места:

Появилась новая строка для ввода

« Примечание: нажатие на пустое пространство в селекторе `element.style` даст возможность записывать CSS-правила прямо в `style=""`-атрибут тега:

Правила можно записывать в тег

Если свойство или его значение введены неверно, браузер оповестит об ошибке:

Индикатор свойства с ошибкой

Если полностью удалить свойство, оно будет стёрто целиком без возможности восстановления. Свойство можно восстановить, если обновить страницу (если нужно скрыть действие свойства, нажмите галочку).

Нажатие на любой цвет или на эффект открывает соответствующее окно с возможностью регулировки параметров:

Иконки для перехода в режим редактирования

Нажатие на цвет:

Редактирование цвета

Нажатие на эффект:

Редактирование эффекта

Наследование свойств

Посмотрим на свойства выбранного DOM-узла и заглянем в `body`:

Список свойств

Браузер пишет, что данные свойства унаследованы от селектора `body` (Inherited from body). Полупрозрачными выделены те свойства, которые не наследуются выбранным DOM-элементом, а непрозрачные — те, которые явно наследуются.

Наследуемые свойства могут быть переопределены, и в этом случае свойство будет зачёркнуто.

Псевдоклассы и псевдоэлементы

Если выбрать нужный псевдокласс в меню `:hov` или указать его через меню `force state`, то параметры псевдокласса можно будет увидеть и изменить:

Отображение псевдокласса

Если для элемента создан псевдоэлемент, то его можно выбрать в DOM-дереве и также изменить стили для него:

Отображение псевдоэлемента

Прочитали главу?

Нажмите кнопку «Готово», чтобы сохранить прогресс.

Готово

⚠ Если вы обнаружили ошибку или неработающую ссылку, выделите ее и нажмите Ctrl + Enter

Поиск по материалам

Git

[Все материалы](#)

В самом начале



- ☐ [Пройдите опрос](#)
- ☐ [Укажите персональные данные](#)
- ☐ [Изучите регламент](#)
- ☐ [Прочитайте FAQ](#)
- ☐ [Добавьте свой Гитхаб](#)
- ☐ [Выберите наставника](#)
- ☐ [Создайте проект](#)

Мой наставник



[Выбрать наставника](#)

Работа с наставником

У вас осталось **10** из 10 консультаций.

[История](#)



Практикум

Тренажёры

Подписка

Для команд и компаний

Учебник по PHP

Курсы

HTML и CSS. Профессиональная вёрстка сайтов

HTML и CSS. Адаптивная вёрстка и автоматизация

JavaScript. Профессиональная разработка веб-интерфейсов

JavaScript. Архитектура клиентских приложений

React. Разработка сложных клиентских приложений

Node.js. Профессиональная разработка REST API

Node.js и Nest.js. Микросервисная архитектура

TypeScript. Теория типов

Алгоритмы и структуры данных

Паттерны проектирования

Webpack

Vue.js 3. Разработка клиентских приложений

Git и GitHub

Анимация для фронтендеров

Блог

С чего начать

Шпаргалки для разработчиков

Отчеты о курсах

Информация

Об Академии

О центре карьеры

Соглашение

Конфиденциальность

Сведения об образовательной организации

Лицензия № 4696

Профессии

Фронтенд-разработчик

JavaScript-разработчик

Фулстек-разработчик

Услуги

Работа наставником

Для учителей

Стать автором

Остальное

Написать нам

Мероприятия

Форум

