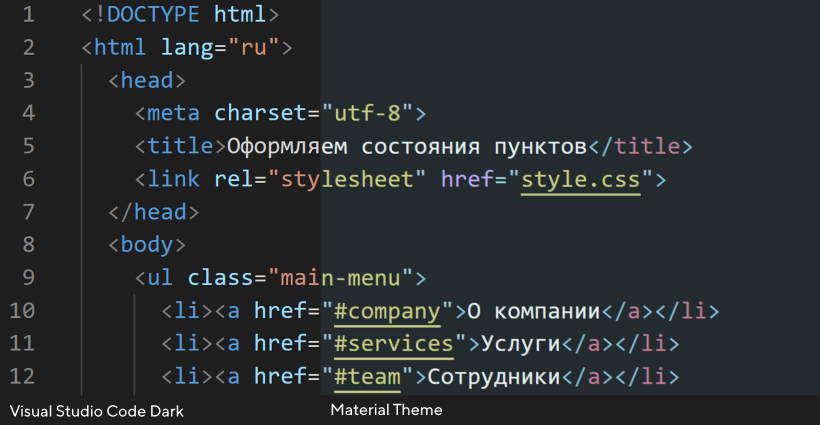
Полезные плагины VS Code для вёрстки

## Emmet

Помогает ускорить вёрстку. С помощью ! и tab можно создать базовую структуру кода за 1 секунду, либо быстро создать вложенные теги.

## Material Theme

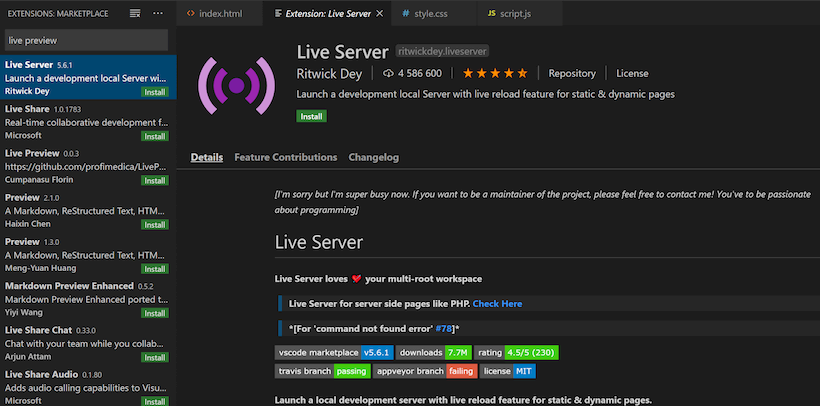
Приятная тема для редактора с разными акцентными цветами.



## 

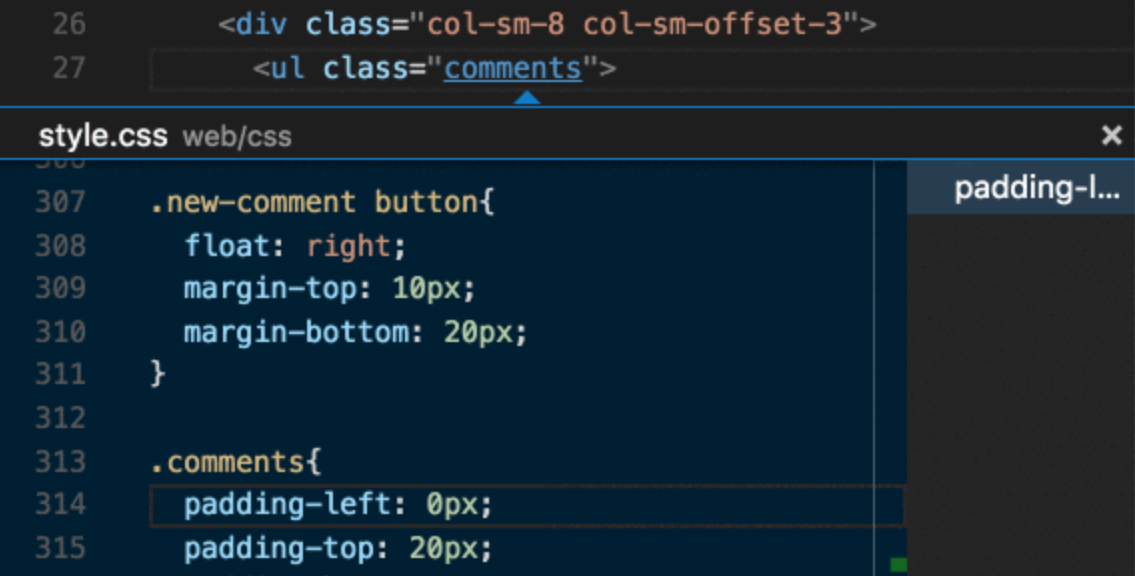
## Live Server

Чтобы сразу увидеть результат работы. Нужно навести на строчку кода, нажать на правую кнопку мыши, выбрать «Open with Live Server» и в браузере отобразится страница, которая автоматически будет перезагружаться. Любые изменения будут отображаться без обновления страницы. Не забывайте отключать плагин с помощью «Stop Live Server»



## CSS Peek

Показывает всплывающее окошко с CSS во время работы над HTML. Это упрощает работу и позволяет не искать нужное свойство среди множества классов в большом CSS-файле.



## Prettier — code formatter

Приводит любой код в нормальный вид за считанные секунды. Комбинация Ctrl+Shift+P отформатирует документ и задаст структуру коду.

## Auto rename tag -

Удобен для переименования тегов в HTML. Если тег повторяется, при замене одного с помощью плагина, все другие будут переименованы автоматически.

Auto Close Tag -

автоматически добавляет закрывающий HTML/XML тег при вводе закрывающей скобки открывающегося тега (кроме HTML, это дополнение добавляет эту возможность в JavaScript и многие другие языки);

## Path autocomplete -

Показывает возможный путь к файлу в кавычках. Не нужно искать папку вручную, путь будет предложен автоматически.

## HTML CSS Support -

Если вы используете много классов и тегов, могут возникнуть ошибки при их написании. Плагин автоматически дополняет название ID или HTML-атрибута для определений, найденных в рабочей области, на которые ссылается link.

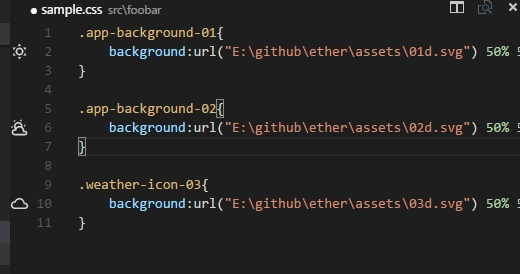
IntelliSense for CSS class names in HTML - плагин для предложения вариантов завершения имени CSS класса в HTML на основе определений, найденных им в вашем рабочем пространстве;

Autoprefixer - для автоматического добавления CSS свойств с префиксами;

Prettier - Code formatter - для форматирования кода (HTML, CSS, JavaScript и др.);

## Image Preview -

Если в вашем коде уже присутствуют ссылки на какие-то изображения/иконки, то [данный плагин](https://timeweb.com/go?url=https%3A%2F%2Fmarketplace.visualstudio.com%2Fitems%3FitemName%3Dkisstkondoros.vscode-gutter-preview&hash=f912c1f7439f62c814a110424f4199cf993e4023) покажет вам их в маленьком окошке слева от порядкового номера строки. Пригодится верстальщикам, чтобы не путаться во множестве иконок.



# 

# VS Code для разработки на JavaScript

## Стандарты кодирования

Eslint – это утилита, проверяющая стандарты кодирования на JavaScript.. Он будет показывать вам ошибки в коде, а также направлять вас при его написании так, чтобы он был выдержан в едином стиле.

Нужно сначала установить eslint в системе, а потом установить расширение VS Code, которое будет использовать установленный линтер. Есть разные способы интеграции линтера с расширением. Мы рассмотрим установку линтера глобально в системе.

1. Установите Node.js, используя [пакетный менеджер вашей операционной системы](https://nodejs.org/en/download/package-manager/).
2. Установите eslint командой

npm install -g eslint

1. Установите плагины, которые конфигурируют eslint. Без них (по умолчанию) eslint ничего не проверяет.

npm install -g eslint-config-airbnb-base eslint-plugin-import

1. eslint требует наличия конфигурационного файла. Создайте в корне вашего проекта файл .eslintrc.yml со следующим содержанием:

extends:

- 'airbnb-base'

env:

node: true

browser: true

1. Установите расширение “[linter-eslint](https://marketplace.visualstudio.com/items?itemName=dbaeumer.vscode-eslint)” в VS Code.

## 

## Автоматическое дополнение

VS Code содержит мощную систему анализа кода для автодополнений и подсказок – [IntelliSense](https://code.visualstudio.com/docs/editor/intellisense).

## C**ode runner**

**Справочник настроек**

Изменение настроек в VS Code осуществляется в соответствующем окне. Открыть его можно несколькими способами:

* через комбинацию клавиш Ctrl+,;
* через пункт меню «Файл -> Настройки -> Параметры» (в версии на английском языке «File -> Preferences -> Settings»);
* нажать на значок шестерёнки и выбрать в открывшемся меню пункт «Параметры» (Settings).

Список параметров, которые пользователи наиболее часто настраивают:

* editor.tabsize - число пробелов при табуляции;
* editor.insertSpaces - вставлять ли пробелы при нажатии Tab;
* editor.detectIndentation - нужно ли параметры "#editor.tabsize" и "editor.insertSpaces" определять автоматически при открытии файла на основе его содержимого;
* editor.wordWrap - управляет тем, как следует переносить строки;
* editor.fontSize - размер шрифта в пикселях;
* editor.mouseWheelZoom - нужно ли включать изменение размера шрифта в редакторе при нажатой клавише Ctrl и движении колесика мыши;
* editor.minimap.enabled - включает или отключает отображение мини-карты;
* editor.formatOnSave - выполнять ли автоматическое форматирование файла при его сохранении;
* workbench.startupEditor - управляет тем, что будет отображаться при запуске, если содержимое редактора не было восстановлено из предыдущего сеанса;
* files.insertFinalNewline - если этот параметр включен, то при сохранении файла в его конец вставляется пустая строка;
* files.trimFinalNewlines - если этот параметр активен, то при сохранении файла будут удалены все пустые строки, идущие за последней в конце файла;
* files.trimTrailingWhitespace - если этот параметр включен, то при сохранении файла будут удалены все пробельные символы на концах строк;
* files.autoSave - для включения автосохранения файлов;
* terminal.integrated.cwd - позволяет задать путь явного запуска, по которому будет запущен терминал;
* telemetry.enableTelemetry - включает или отключает отправку сведений об использовании и ошибках в веб-службу Майкрософт;
* telemetry.enableCrashReporter - разрешает отправку отчетов о сбоях в веб-службу Майкрософт;