

Редакторы кода

Содержание

Редакторы кода и IDE	3
1. Sublime Text 3	4-12
◦ Преимущества	5
◦ Используйте лучшие плагины	6
◦ Создавайте сниппеты	8
◦ Пользуйтесь GitHub Gist	10
2. PhpStorm	13-23
◦ О PhpStorm	14
◦ PhpStorm и WebStorm. В чём отличие	14
◦ Основные возможности	14
◦ Установка тем оформления	15
◦ Как включить нумерацию строк	17
◦ Регулировка размера шрифта колесиком мыши	17
◦ Установка типа перевода строки	18
◦ Настройка терминала (для Windows)	18
◦ Создание сниппетов	19
◦ Работа с GitHub Gist	22
◦ Плагин WebFont	23

Редакторы кода и IDE

За написанием и отладкой кода разработчик ежедневно проводит 80% своего времени. А это 10-14 часов в программе для работы с кодом! Она становится главной магистралью вашего рабочего процесса, частью вас и того, как вы видите свой код и свои проекты. Именно поэтому *выбор хорошего редактора кода или IDE так важен для любого разработчика.*

Лёгкий редактор кода или IDE? Споры на эту тему в среде программистов не умолкают, свои плюсы и минусы есть у всех существующих редакторов и сред разработки.

IDE (Integrated Development Environment) — «интегрированная среда разработки» - это редактор, который расширен большим количеством «наворотов», умеет работать со вспомогательными системами, такими как багтрекер, контроль версий, и др.

Главный плюс IDE - наличие синтаксического анализатора языка программирования. *Среда «понимает», что она редактирует код.*

Редактор — нет. А это и автодополнение, и навигация, и подсветка синтаксических, а, иногда, и семантических ошибок. Если команда или выражение синтаксически не верны, умная IDE не предложит их вам в качестве подсказки.

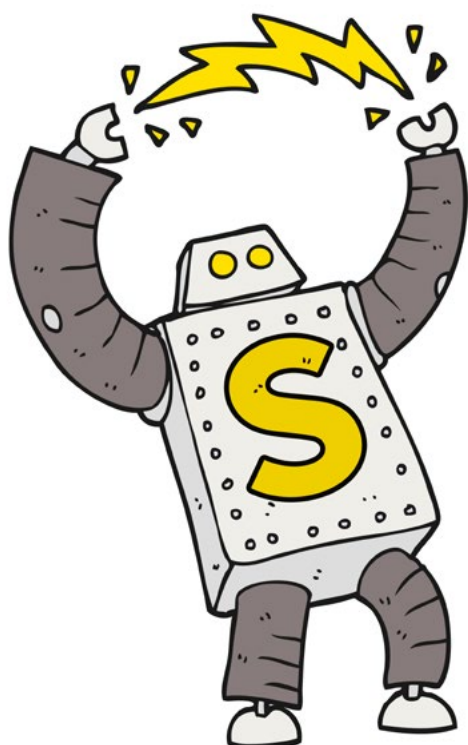


Основное преимущество **лёгких редакторов** - это скорость работы и простота - они идеальны для мелких проектов с небольшим количеством файлов. Редакторы могут работать не только с кодом. Всю их мощь можно задействовать при работе с csv или xml файлами. Либо чего-то другого, в чём иногда возникает необходимость, вроде статьи или письма. Сейчас для лёгких редакторов существует огромное количество плагинов, поэтому граница между ними и IDE очень размыта. А удовольствие от непрерывного тюнинга своего основного инструмента и ощущение полного контроля над ним - никто не отменял!



1

SUBLIME TEXT 3



Sublime Text 3 — это мощный и красивый редактор кода. Этот редактор славится своей быстрой работой (так как написан на C++), достаточно развитым функционалом и наличием огромного количества плагинов, позволяющих сделать из него удобную, заточенную практически под любой современный язык программирования IDE.



[Загрузить Sublime Text 3](#)

[Документация на русском языке](#)

[Горячие клавиши Windows](#)

[Горячие клавиши OSX](#)

[Классная презентация](#)

[Видеоурок от Дмитрия Ковальчука](#)

[Запуск Sublime Text из терминала Mac OS X](#)

Преимущества

- Скорость работы
- Минималистичный интерфейс — ничего лишнего
- Графическая карта файла — удобная идея навигации по файлу
- Нечёткий поиск
- Гибкая настройка под себя без графического интерфейса
- Мощное, развитое сообщество плагинов, обеспеченное первоначальником Package Control (1000+ плагинов от 800+ авторов)
- Мультикурсор — редактирование столбцов текста и не только
- Продуманные хоткеи
- Таб-группы — возможностью разделить экран на несколько частей
- Возможность одновременно редактировать один и тот же файл в разных табках.

Используйте лучшие плагины

Для установки плагинов необходимо загрузить ***Package Control*** — менеджер плагинов Sublime Text 3, позволяющий моментально устанавливать, удалять и обновлять их.

1. Вызовите консоль Sublime Text 3, набрав `ctrl+`` или в меню `View > Show Console`.
2. Вставьте в консоль этот текст:

```
import urllib.request,os,hashlib; h =
'2915d1851351e5ee549c20394736b442' +
'8bc59f460fa1548d1514676163dafc88'; pf = 'Package
Control.sublime-package'; ipp =
sublime.installed_packages_path();
urllib.request.install_opener( urllib.request.build_opener(
urllib.request.ProxyHandler()) ); by = urllib.request.urlopen(
'http://packagecontrol.io/' + pf.replace(' ', '%20')).read(); dh
= hashlib.sha256(by).hexdigest(); print('Error validating
download (got %s instead of %s), please try manual install' %
(dh, h)) if dh != h else open(os.path.join( ipp, pf), 'wb'
).write(by)
```

3. Для установки плагинов вызовите Package Control (`CTRL/⌘-SHIFT-P`), выберите `Install Package`, в загрузившемся списке плагинов найдите нужный и... готово!

Emmet

Один из самых популярных для верстальщика плагинов, который позволит вам писать html и css в десятки раз быстрее.

Справочник ЕММЕТ-сокращений

SideBarEnhancements

Если вы кликните правой кнопкой мыши по какому-то файлу сайдбара, обнаружите, что появляющееся при этом контекстное меню довольно скромное. Функционал этого меню можно значительно расширить, установив плагин SideBarEnhancements, после чего возможности контекстного меню значительно расширятся.

AdvancedNewFile

Быстро создаем новые папки и файлы с помощью горячих клавиш.

BracketHighlighter

Удобная подсветка открывающих и закрывающих скобок в коде.

ZenTabs

Подсвечивает текущую вкладку, ограничивает количество вкладок (задаем нужное количество сами). Когда количество вкладок больше заданного, закрывает самые старые.

SublimeLinter

Позволяет делать проверку кода на наличие синтаксических ошибок. Линтер читает код и подсвечивает ошибки и предупреждения, при наведении на которые объясняет ошибку в нижней строчке. Для каждого языка необходимо установить свой линтер.

Gutter Color

Полноценный color picker прямо в окне вашего редактора.

AutoFileName

Позволит вам не вспоминать полные названия файлов и пути до них, когда вы будете подключать их в коде.

jQuery

Автокомплит и разные дополнения для jQuery

Sublime SFTP

Два режима работы с FTP: 1) подключение к удаленному серверу и работа на нем; 2) настройка FTP-соединения для каждого проекта, позволяющая быстро и просто синхронизировать файлы.

Gist

Позволяет сохранять отдельные участки кода на github прямо во время редактирования.

Чтобы посмотреть список всех установленных плагинов, вызовите Package Control (CTRL+⌘-SHIFT-P), выберите List Packages

Создавайте сниппеты

Сниппеты — это небольшие куски кода, которые можно вставлять по определенному сочетанию клавиш. Разработчики на протяжении всего процесса кодирования снова и снова используют одни и те же куски кода. Одним из способов устранить этот повторяющийся процесс, является, как раз-таки, использование сниппетов.

Если вы каждый раз прописываете команду `console.log()` вручную, эта глава для вас. Давайте сделаем сниппет. Чтобы создать новый сниппет, переходим *Tools - Developer - New Snippet*.

Что всё это значит? Давайте разбираться.

```
1 <snippet>
2   <content><![CDATA[
3 Hello, ${1:this} is a ${2:snippet}.
4 ]]></content>
5   <!-- Optional: Set a tabTrigger to define how to trigger the snippet -->
6   <!-- <tabTrigger>hello</tabTrigger> -->
7   <!-- Optional: Set a scope to limit where the snippet will trigger -->
8   <!-- <scope>source.python</scope> -->
9 </snippet>
```

<tabTrigger>hello</tabTrigger> Здесь прописывается триггер для сниппета. То есть, если мы введем *hello* и нажмём *Tab*, сниппет сработает.

<scope>source.python</scope> Содержимое этой строки означает, что сниппет будет работать в скриптах python. Чтобы создать сниппет, который будет работать в html, css или javascript необходимо использовать другие значения для строки `<scope>`.

Полный список всех возможных значений строки `<scope>`

Если оставить строку `<scope>` закомментированной — то этот сниппет будет работать во всех файлах.

Рассмотрим блок с тегом `<content>`. В параметре `CDATA[...]` указан собственно текст сниппета, то что будет выведено когда мы наберем *hello* (значение `tabTrigger` в нашем примере) и нажмем TAB.

`${1:this}` и `${2:snippet}` - это ключевые точки для редактирования сниппета. После вызова сниппета, перемещаться по ним можно используя клавишу TAB. В данном примере ключевые точки заданы с некоторыми значениями по умолчанию. Можно задать свое значение по умолчанию, которое выделится для редактирования, когда мы создадим сниппет, либо сделать пустую ключевую точку. Создавать пустую ключевую точку имеет смысл тогда когда просто необходимо поставить курсор в нужное место.

Итак, давайте создадим свой сниппет:

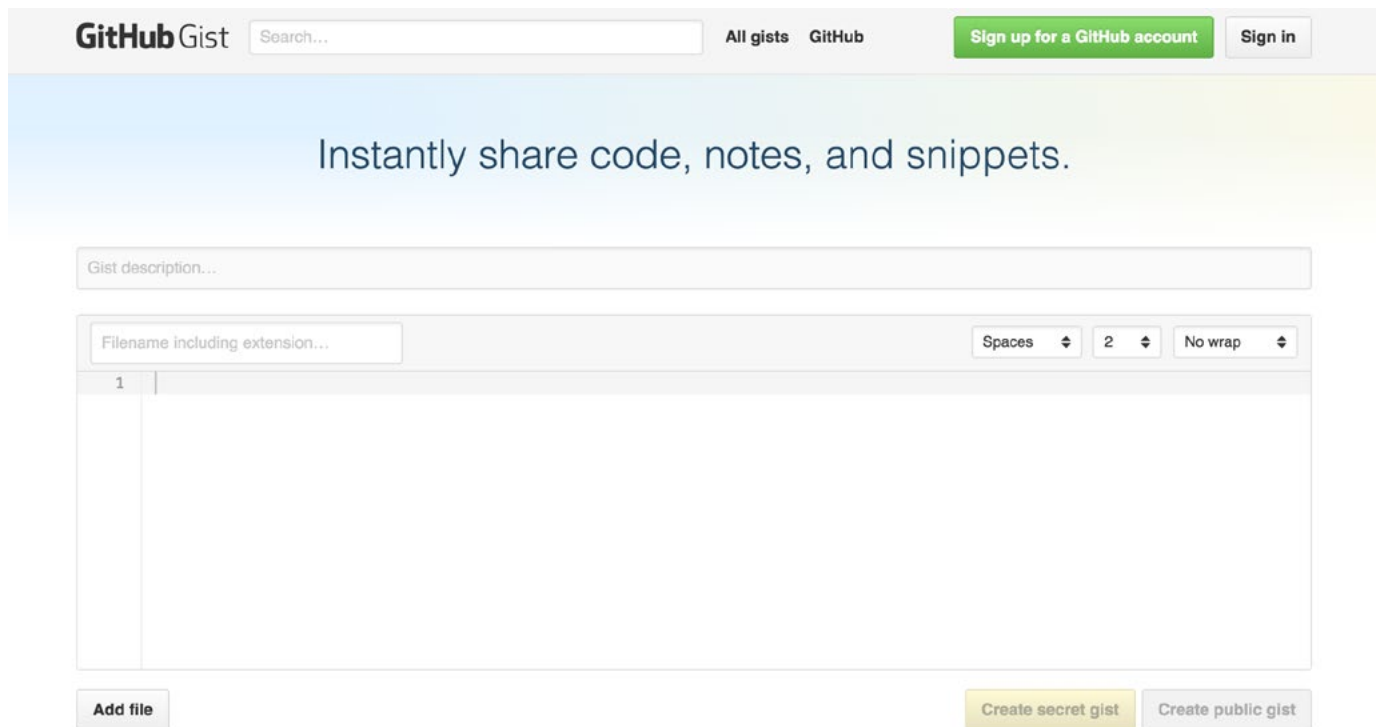
```
1 <snippet>
2   <content><![CDATA[
3     console.log($1);$2
4   ]]></content>
5   <tabTrigger>cons</tabTrigger>
6   <scope>source.js</scope>
7 </snippet>
```

Сохранять файл со сниппетом можно в папку *Packages/User*. По умолчанию Sublime предлагает сохранить его под именем *untitled.sublime-snippet*. Имя файла изменяем на свое усмотрение, а *расширение необходимо оставить неизменным*.

Теперь, если в файле *js* мы введём *cons* и нажмем *Tab*, сработает наш сниппет. Все просто!

Пользуйтесь GitHub Gist

Где вы храните блоки кода, которые повторяются и копируются из проекта в проект? **GitHub Gists** — удобный и бесплатный сервис для хранения кода от GitHub, огромным плюсом которого является интеграция с Sublime Text 3. Можно сохранять любой блок кода с помощью одной комбинации клавиш и также быстро добавлять сохраненный код в новом проекте.



The screenshot shows the GitHub Gist web interface. At the top, there's a header with the 'GitHub Gist' logo, a search bar, and links for 'All gists', 'GitHub', 'Sign up for a GitHub account', and 'Sign in'. Below the header is a light blue banner with the text 'Instantly share code, notes, and snippets.' The main form area includes a 'Gist description...' text box. Below that is a 'Filename including extension...' input field, followed by dropdown menus for 'Spaces' (set to 2), '2' (likely line count), and 'No wrap'. A large text area for the code is below these, with a line number '1' visible on the left. At the bottom of the form, there are three buttons: 'Add file', 'Create secret gist', and 'Create public gist'.

В первую очередь, установим плагин **Gist** для Sublime Text 3. Про установку плагинов рассказывается чуть выше.

Для того, чтобы привязать наш аккаунт на GitHub, необходимо **сгенерировать токен**:

- В меню своего профиля на GitHub выберите пункт *Settings* и вкладку *Personal access tokens*.
- Кликните на кнопку «*Create new token*».
- В открывшемся окне нам нужно ввести лишь *имя* (например, *Sublime Text*) и поставить галочку напротив пункта *Gist*. Генерируем токен.

Token description

Sublime Text

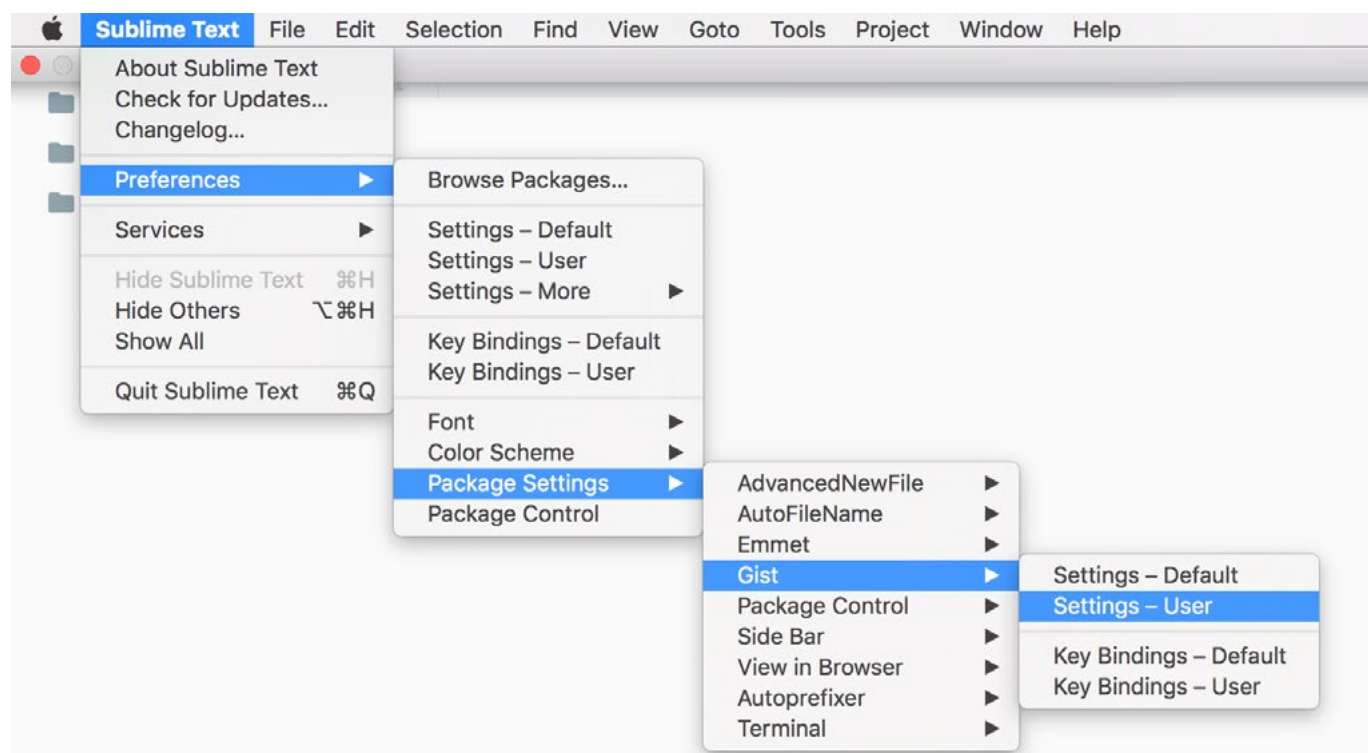
What's this token for?

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> admin:org	Full control of orgs and teams
<input type="checkbox"/> write:org	Read and write org and team membership
<input type="checkbox"/> read:org	Read org and team membership
<input type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input type="checkbox"/> write:repo_hook	Write repository hooks
<input type="checkbox"/> read:repo_hook	Read repository hooks
<input type="checkbox"/> admin:org_hook	Full control of organization hooks
<input checked="" type="checkbox"/> gist	Create gists
<input type="checkbox"/> notifications	Access notifications

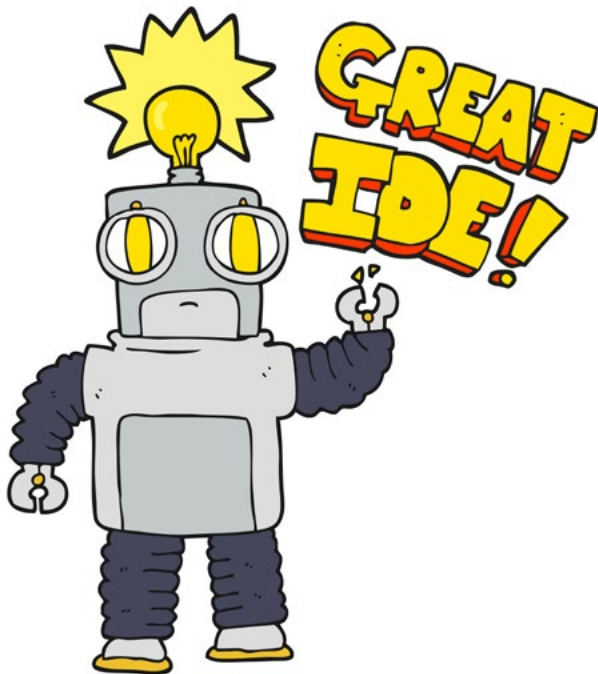
Полученный токен необходимо добавить в файл настроек, параметру *token*, а для параметра *include_users* указать ваш логин для GitHub. Сохраняемся.



Чтобы использовать сохраненный ранее gist, нужно нажать *Ctrl+K+O*. Для сохранения блока кода в новый gist выделите код и нажмите *Ctrl+K+I*, после - введите имя гиста и сохраните его.

2

PhpStorm



О PhpStorm

PhpStorm — это интегрированная среда разработки на PHP с интеллектуальным редактором, которая глубоко понимает код и содержит полный набор инструментов для фронтенд-разработки.

PhpStorm и WebStorm. В чём отличие

PhpStorm включает в себя всю функциональность WebStorm (HTML/CSS редактор, JavaScript редактор) и добавляет полнофункциональную поддержку PHP и баз данных / SQL. **WebStorm** же позволяет эффективно разрабатывать приложения на Node.js и поддерживает полноценную отладку Node.js приложений. Новое приложение можно создать, используя шаблон Node.js Express, а необходимые модули установить с помощью встроенного в WebStorm менеджера npm.

Основные возможности

- Интеллектуальный редактор PHP кода с подсветкой синтаксиса, автодополнением кода, расширенными настройками форматирования кода, предотвращением ошибок налету.
- Интеграция с системами управления версиями, включая унифицированный интерфейс.
- Поддерживает PHP 7.0, 5.6, 5.5, 5.4 и 5.3, генераторы, сопрограммы и все синтаксические улучшения.
- PHP рефакторинги, code (re)arranger, детектор дублируемого кода.
- Поддержка Vagrant, Composer, встроенный REST клиент, Command Line Tools, SSH консоль.
- Поддержка фреймворков (MVC view для Symfony2, Yii) и специализированные плагины для ведущих PHP фреймворков (Symfony, Magento, Drupal, Yii, CakePHP и многие другие).

- HTML, CSS, JavaScript редактор. Отладка и модульное тестирование для JS. Поддержка HTML5, CSS, Sass, SCSS, Less, Stylus, Compass, CoffeeScript, TypeScript, ECMAScript Harmony, Emmet и других передовых технологий веб-разработки.
- Удаленное развертывание приложений и автоматическая синхронизация с использованием FTP, SFTP, FTPS и др.
- Live Edit: изменения в коде можно мгновенно просмотреть в браузере без перезагрузки страницы.
- Инструменты работы с базами данных, SQL редактор.

[Страница загрузки](#)

[Список горячих клавиш](#) для каждой ОС в удобном pdf-формате (англ.)

Установка тем оформления

Скачать темы можно здесь:

[color-themes.com](#)

[phpstorm-themes.com](#)

[Material Theme](#)

Все настройки IDE находятся в меню PhpStorm → Preferences (OSX/Unix: \mathbb{H} + , Windows: Ctrl+Alt+S)

Чтобы *импортировать скачанную тему*, перейдем в меню File → Import Settings и выберем файл темы. Перезапускаем программу.

Установить так популярную сейчас *Material Theme* можно через меню *PhpStorm* → *Preferences* → *Plugins*. Выберите *Browse repositories* и введите в поле поиска *Material Theme UI*. Установите тему.

```
new Vue({
  el: '#clubs',

  data: {
    searchClub: { club_id: '' },
    athletes: [],
    searching: false
  },

  methods: {
    fetchAthletes: function(club_id) {
      var self = this;

      this.athletes = [];
      this.searching = true;

      this.$http.get('api/leaderboard/' + club_id, function(athletes) {
        this.searching = false;

        self.athletes = athletes;
      });
    },

    onsubmitForm: function(e) {
      e.preventDefault();

      var clubid = this.searchClub.club_id;

      this.fetchAthletes(clubid);
    }
  }
})
```

```
new Vue({
  el: '#clubs',

  data: {
    searchClub: { club_id: '' },
    athletes: [],
    searching: false
  },

  methods: {
    fetchAthletes: function(club_id) {
      var self = this;

      this.athletes = [];
      this.searching = true;

      this.$http.get('api/leaderboard/' + club_id, function(athletes) {
        this.searching = false;

        self.athletes = athletes;
      });
    },

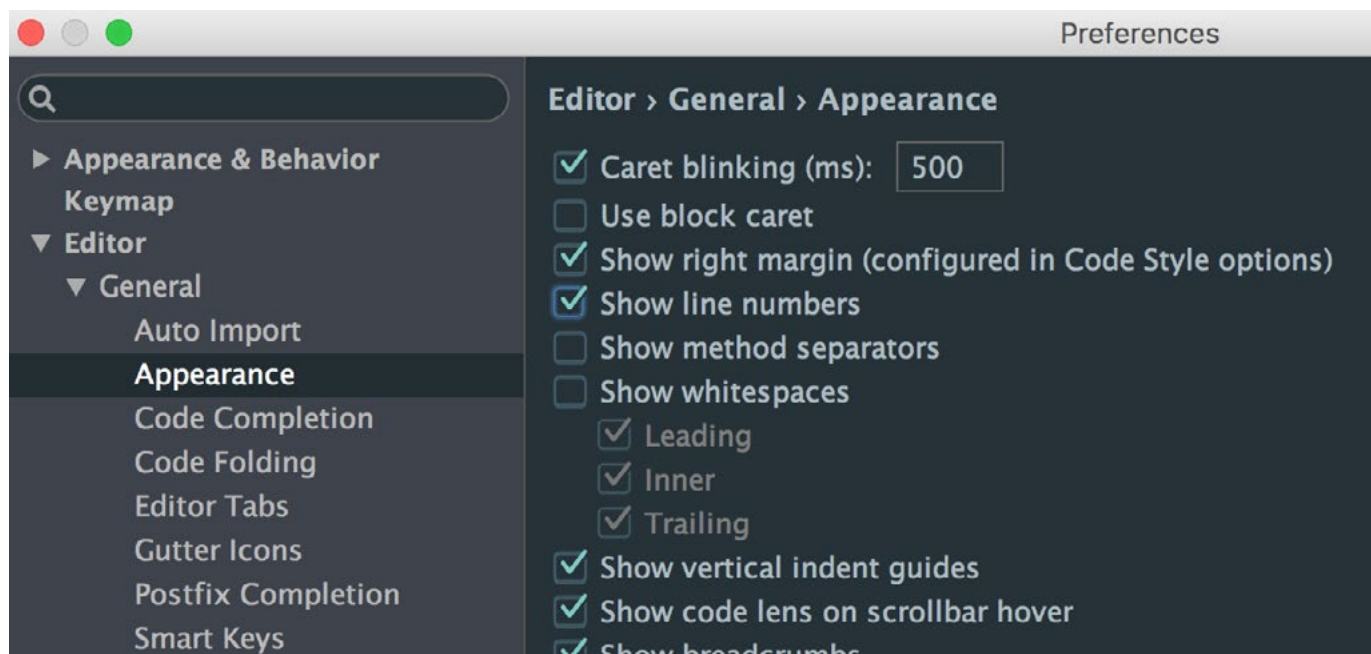
    onsubmitForm: function(e) {
      e.preventDefault();

      var clubid = this.searchClub.club_id;

      this.fetchAthletes(clubid);
    }
  }
})
```

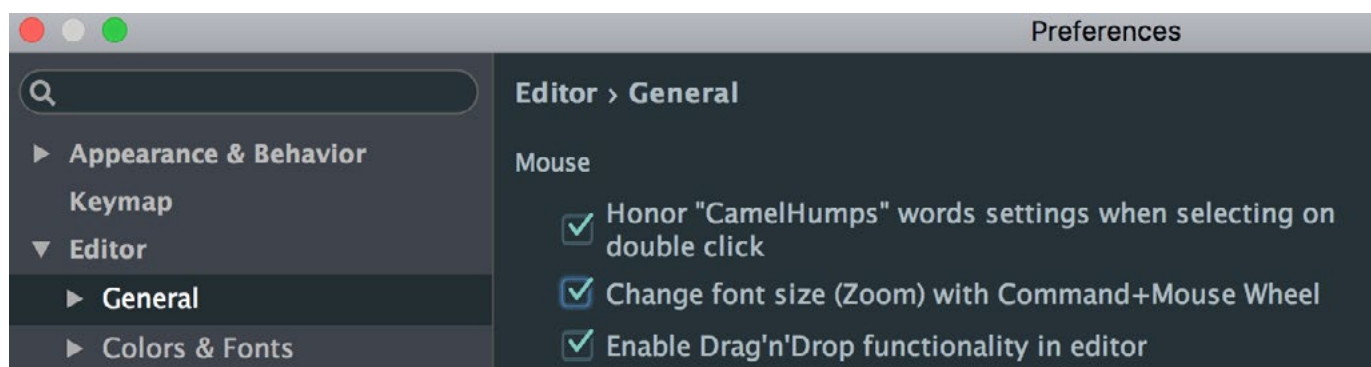

Как включить нумерацию строк

Нумерация строк отключена по умолчанию. Чтобы ее включить, перейдем в меню *PhpStorm* → *Preferences* → *Editor* → *General* → *Appearance* и поставим галочку в строке *Show line number*



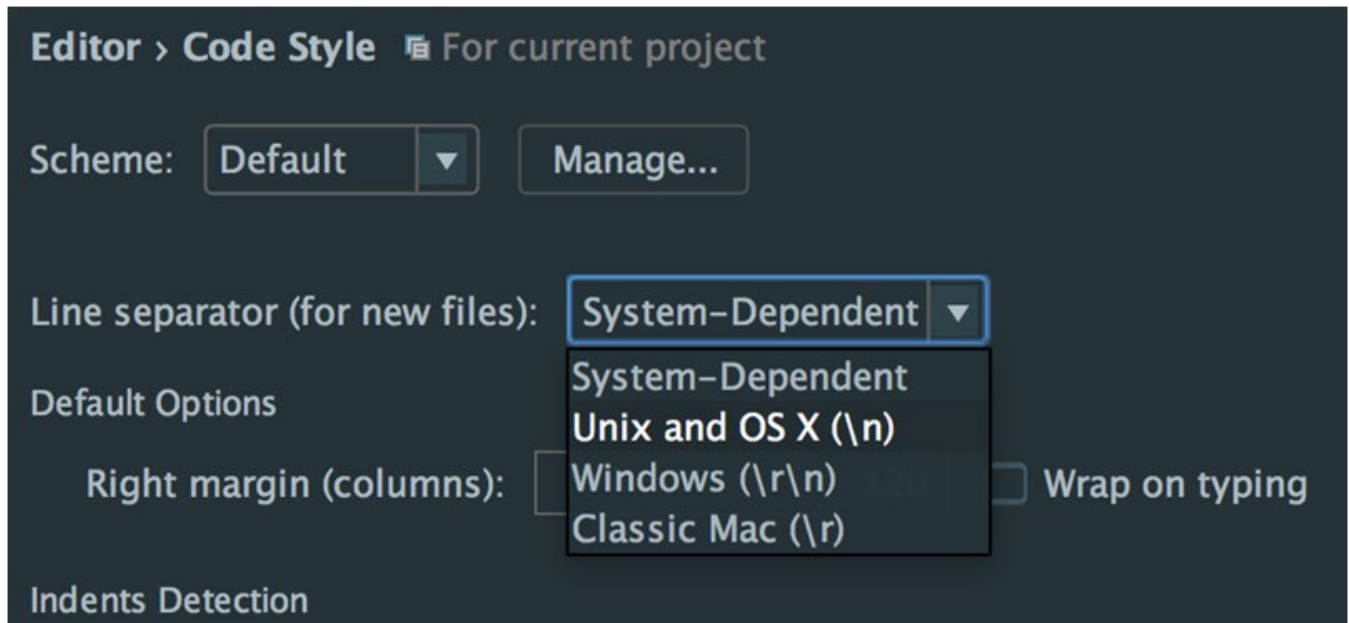
Регулировка размера шрифта колесиком мыши

Возможность регулировать размер шрифта колесиком мыши и зажатой клавишей ⌘ (или Ctrl) по умолчанию отключена. Чтобы ее включить, перейдем в меню *PhpStorm* → *Preferences* → *Editor* → *General* и поставим галочку в строке *Change font size (Zoom) with Command + Mouse Wheel*



Установка типа перевода строки

PhpStorm позволяет установить тип перевода строки для вновь создаваемых файлов, а также изменить стиль линий разделитель для существующих файлов. Рекомендуем включить Unix тип в меню *PhpStorm* → *Preferences* → *Editor* → *Code Style*:



Настройка терминала (для Windows)

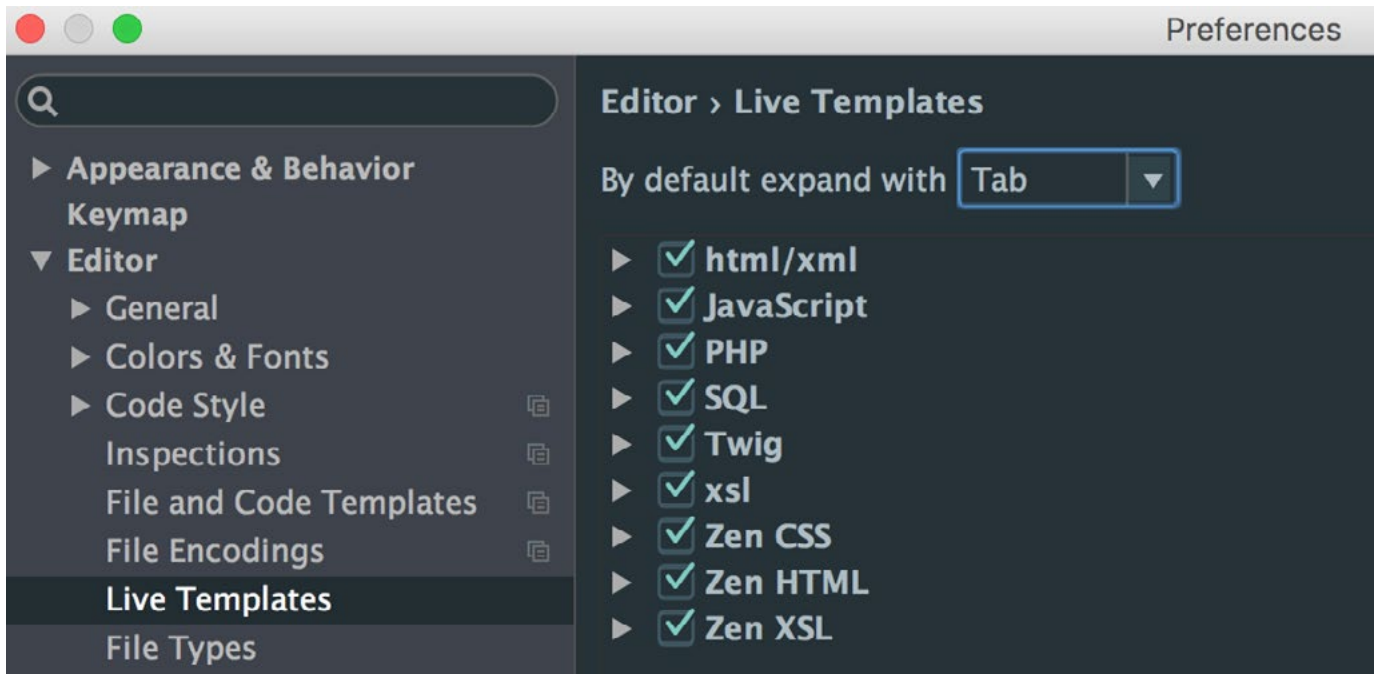
По-умолчанию, терминал в PhpStorm - стандартный терминал Windows cmd.exe. Для того чтобы настроить в PhpStorm другой терминал, например, git bash, перейдем *File* → *Settings* → *Tasks* → *Terminal*

В поле *Shell path* пропишем путь до нужного .exe файла, например:
`C:\Program Files (x86)\Git\bin\sh.exe" -login -i`

-login и -i нужны для загрузки конфигурационных файлов и включения интерактивного режима.

Создание сниппетов

Для создания сниппета перейдем в *PhpStorm* → *Preferences* → *Editor* → *Live Templates*. Сниппеты разбиты на группы, мы можем менять их, удалять и добавлять свои.

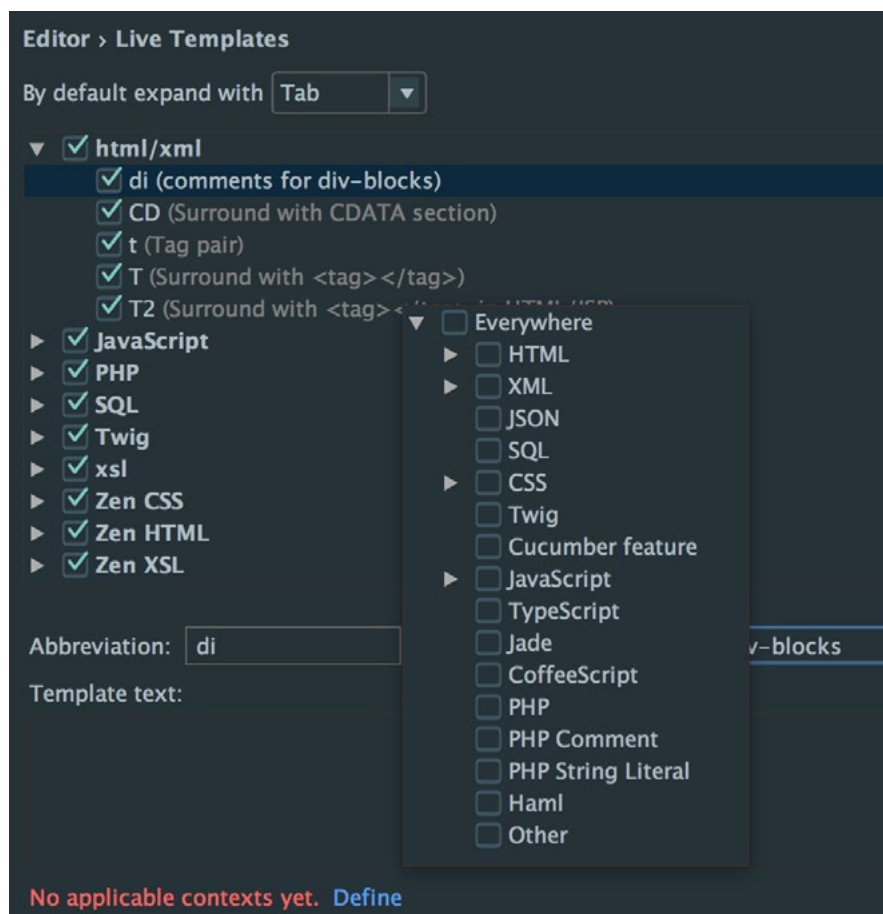


Давайте создадим новый сниппет для html-файлов. Введя триггер *div* и нажав *Tab*, мы сможем обернуть любой div-блок комментариями, благодаря которым легко понять, где он начинается и заканчивается.

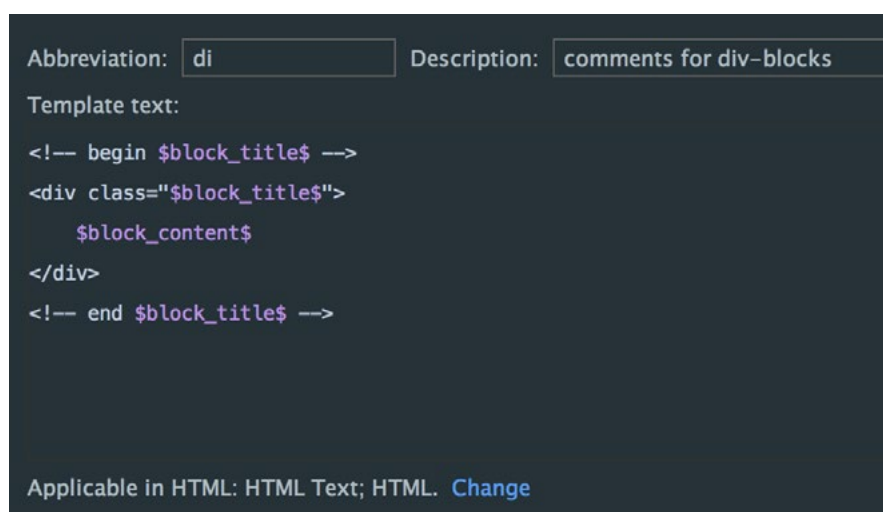
```
1      <!-- begin header -->
2      <div class="header">
3
4      </div>
5      <!-- end header -->
```

Для этого выделим пункт *html/xml*, нажмем значок плюса справа и выберем *Live Template*. В поле *Abbreviation* введем триггер, в нашем случае, это *div*, заполним поле *Description*.

Обратите внимание, на красную надпись внизу окна. Мы должны выбрать контекст, то есть те типы файлов, в которых будет работать наш сниппет. Выберем группу HTML в выпадающем списке.



Введем текст сниппета.



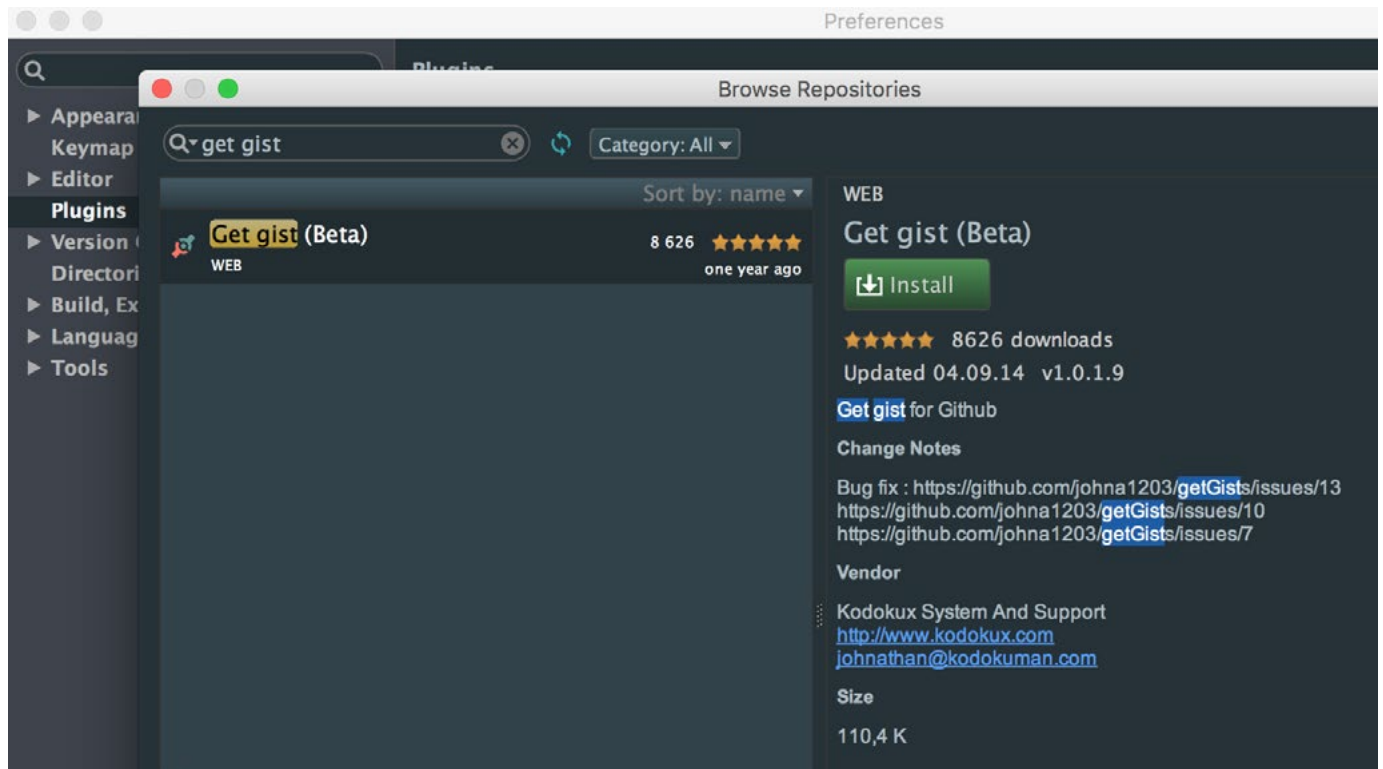
Между знаками доллара после вызова snippets будет установлен курсор. Так как мы ввели одинаковый текст в трех местах, то текст будет набираться одновременно в трех полях. А после нажатия Tab, переместится внутрь div-блока.

```
1      <!-- begin content -->
2      <div class="content">
3          Сниметы – это удобно!
4      </div>
5      <!-- end content -->
```

Снимет в действии (по ссылке - гифка)

Работа с GitHub Gist

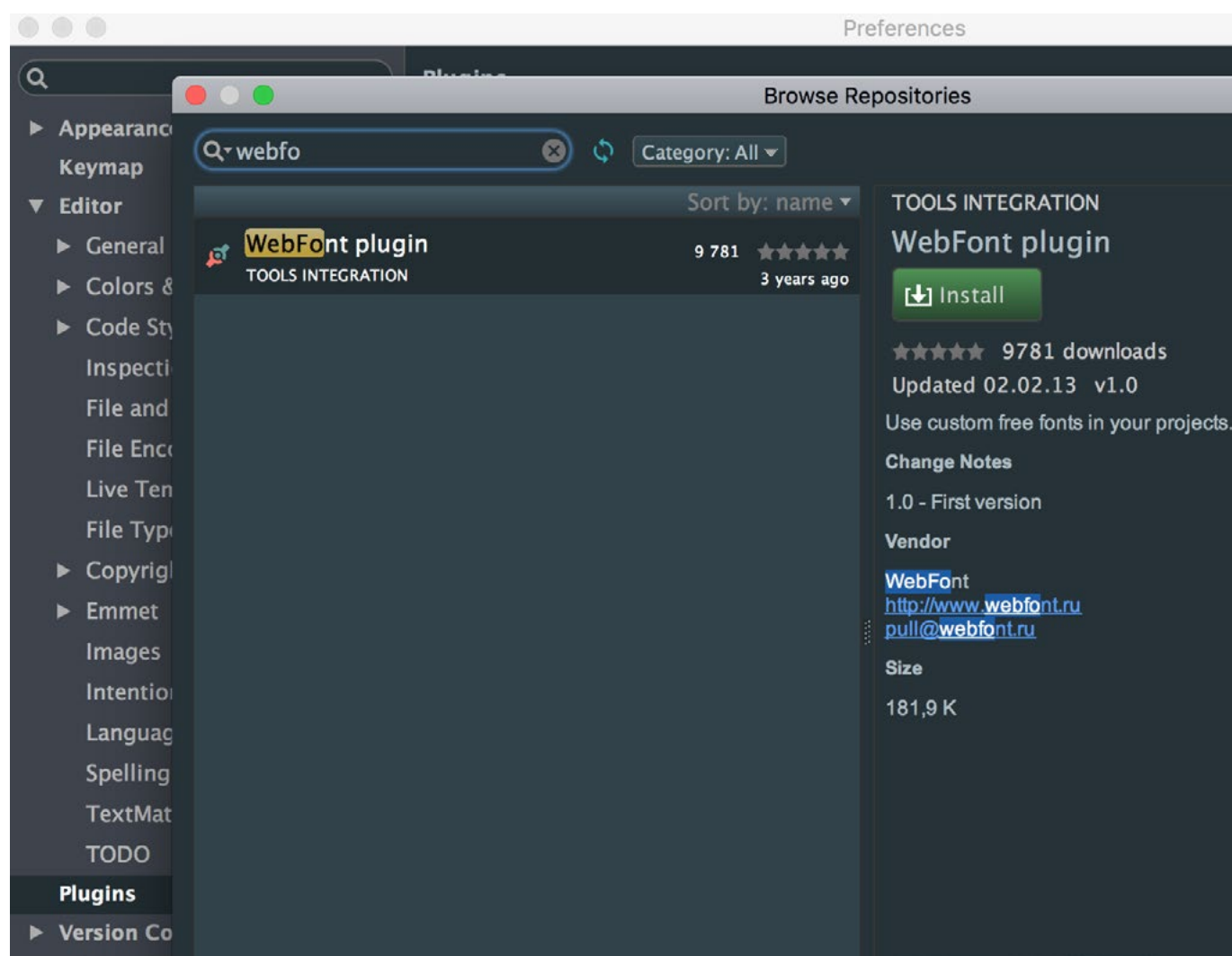
О сервисе **GitHub Gists** уже рассказано выше - в главе про Sublime Text 3. В первую очередь необходимо установить Get gist в меню PhpStorm → Preferences → Plugins. Выберите Browse repositories и введите в поле поиска Get Gist. Устанавливаем и перезапускаем программу.



Для того, чтобы привязать наш аккаунт на GitHub, идем в меню *PhpStorm* → *Preferences* → *Version Control* → *GitHub*, вводим логин и пароль от аккаунта на GitHub. Готово! Чтобы использовать гист, кликаем правой кнопкой мыши на открытом документе, и выбираем пункт *Get Gist*.

Плагин WebFont

Плагин для работы со шрифтами от сервиса WebFont.



После его установки в меню *PhpStorm* появится пункт *WebFont*, в котором есть два пункта: *Import Font* для подключения шрифта с удаленного репозитория и *Download Font* для загрузки шрифта локально.