

PHPStorm

PhpStorm представляет собой интеллектуальный редактор для PHP, HTML и JavaScript с возможностями анализа кода на лету, предотвращения ошибок в коде и автоматизированными средствами рефакторинга для PHP и JavaScript. Автодополнение кода в PhpStorm поддерживает спецификацию PHP 5.3, 5.4, 5.5, 5.6 и 7 (современные и традиционные проекты), включая генераторы, сопрограммы, пространства имен, замыкания, типаж и синтаксис коротких массивов. Имеется полноценный SQL-редактор с возможностью редактирования полученных результатов запросов.

PhpStorm разработан на основе платформы IntelliJ IDEA, написанной на Java. Пользователи могут расширить функциональность среды разработки за счет установки плагинов, разработанных для платформы IntelliJ, или написав собственные плагины.

PhpStorm предоставляет очень гибкую и мощную систему горячих клавиш. Мы можем выбрать несколько предустановленных пресетов клавиш

Подсказки (Code Completion)

Важной функцией IDE является способность подсказывать названия и атрибуты методов. Особенно это важно для ООП, когда цепочки по 3-4 наследующихся класса способны запутать кого угодно.

При нормальной архитектуре проекта IDE способна разобраться сама, но бывают ситуации когда она не справляется, например:

- Функция может вернуть несколько типов
- Объект является элементом массива
- `$var = $arr['object'];`

Класс 'B' extends 'A'. В классе 'A' метод foo() возвращает \$this, в классе 'B' реализация метода отсутствует. Вызвав из объекта 'B' метод foo() IDE будет думать что возвращен был объект A.

При нативной шаблонизации средствами PHP. Где-то в дебрях системы в шаблон передается переменная \$some, но внутри шаблона об этом неизвестно.

Варианты решения:

Для функций нужно писать PHPDoc с параметром @ return. Например:

```
/** @return SomeClass */  
function a() {...}
```

Для переменных также можно указать комментарий:

```
/** @var $my_var SomeClass */
```

Тогда ниже для `$my_var` будут доступны подсказки

Для методов над классом можно указать PHPDoc

```
/** @method B foo() Описание метода */  
class B extends A {...}
```

Если функция возвращает массив объектов

```
/** @return SomeObject[] */
```

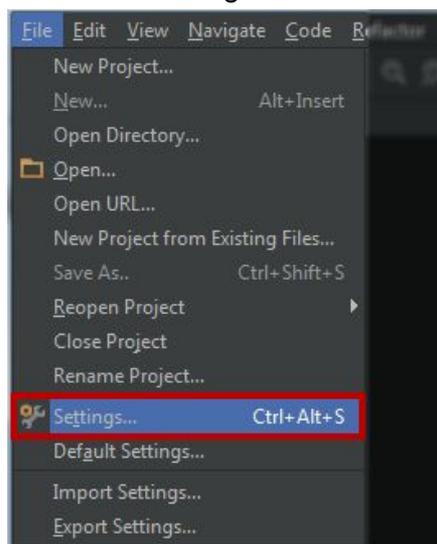
Тогда, например, в `foreach` становятся доступны подсказки `$value->...` (по полям и методам класса `SomeClass`).

Консоль

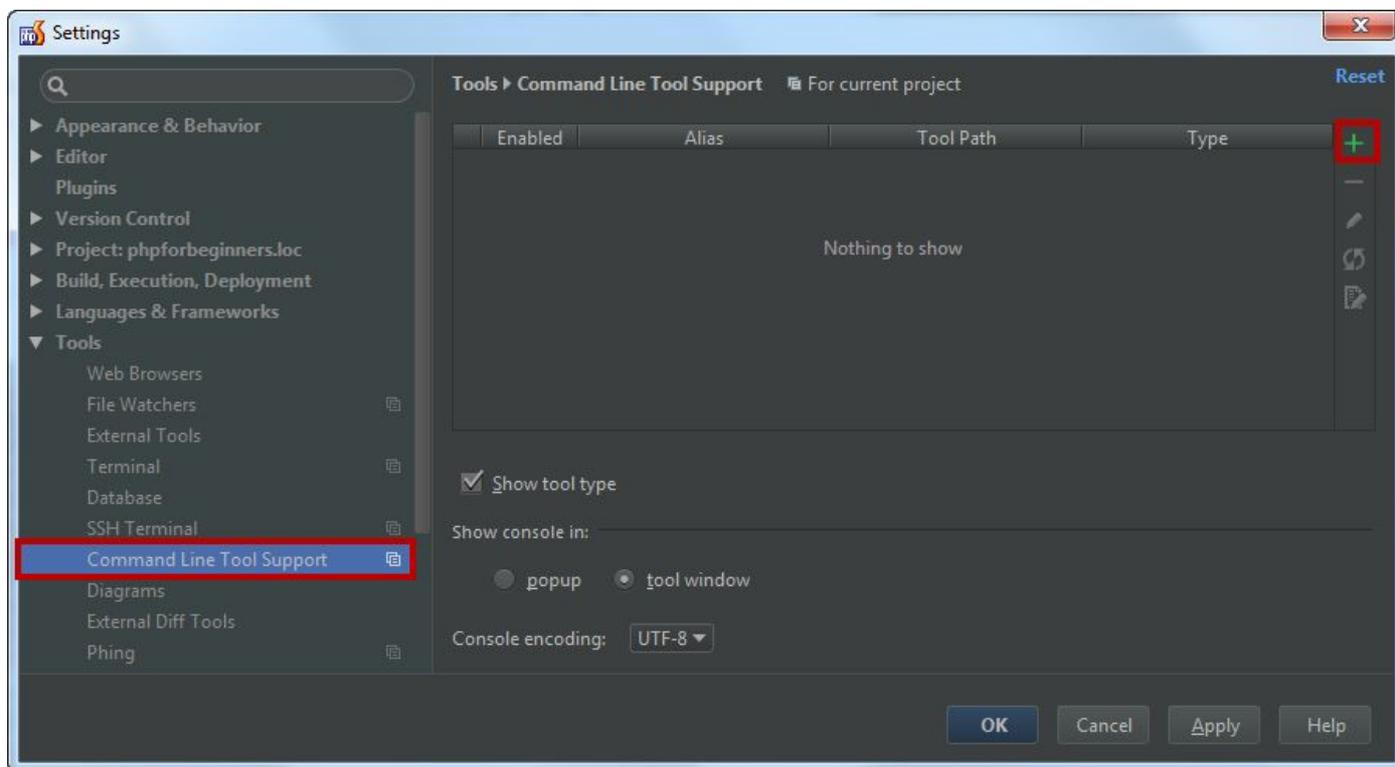
Многие фреймворки используют консоль для быстрого выполнения частых действий. Для `Symfony`, `Zend` и других популярных фреймворков `PHPStorm` поддерживает подсказки и автодополнение консольных команд в `Command Line Tools Console` [CNTRL] + [SHIFT] + X

Для примера, давайте добавим поддержку команд пакетного менеджера `composer` в нашем проекте.

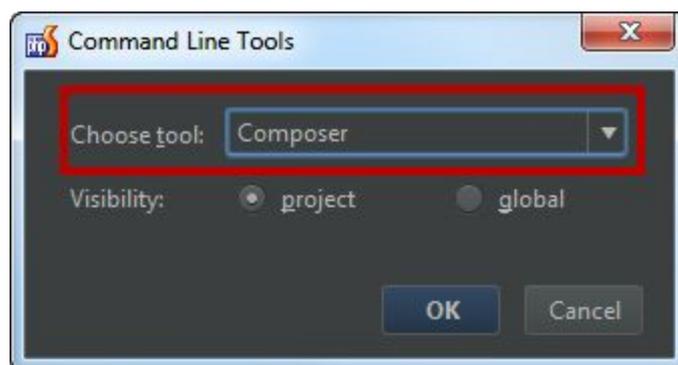
Переходим в настройки проекта `File - Settings`



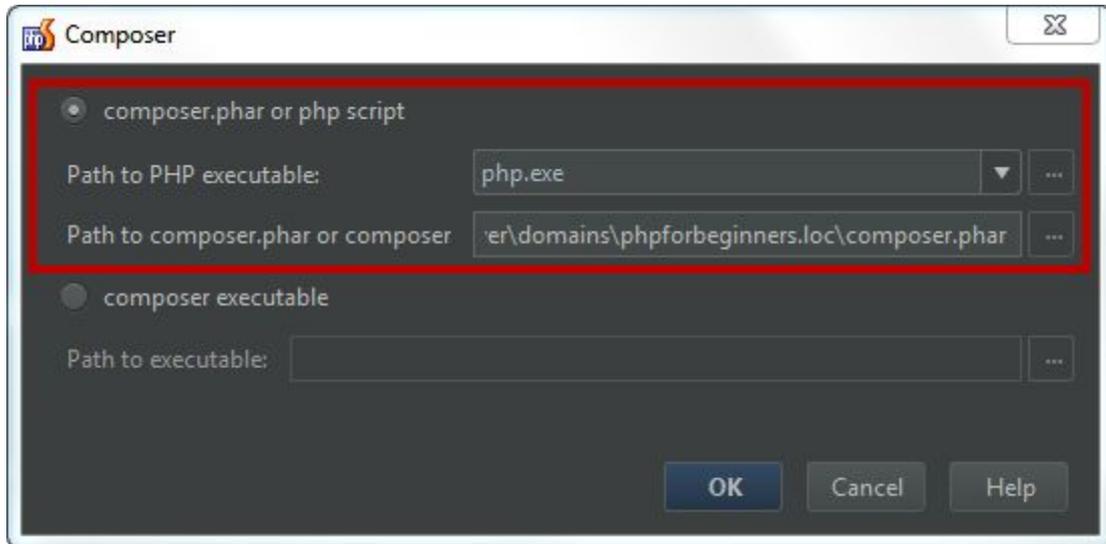
В открывшемся окне выбираем пункт меню "Command Line Tool Support". Далее, в правой части окна нажимаем на кнопку "+" для того, чтобы добавить поддержку пакетного менеджера



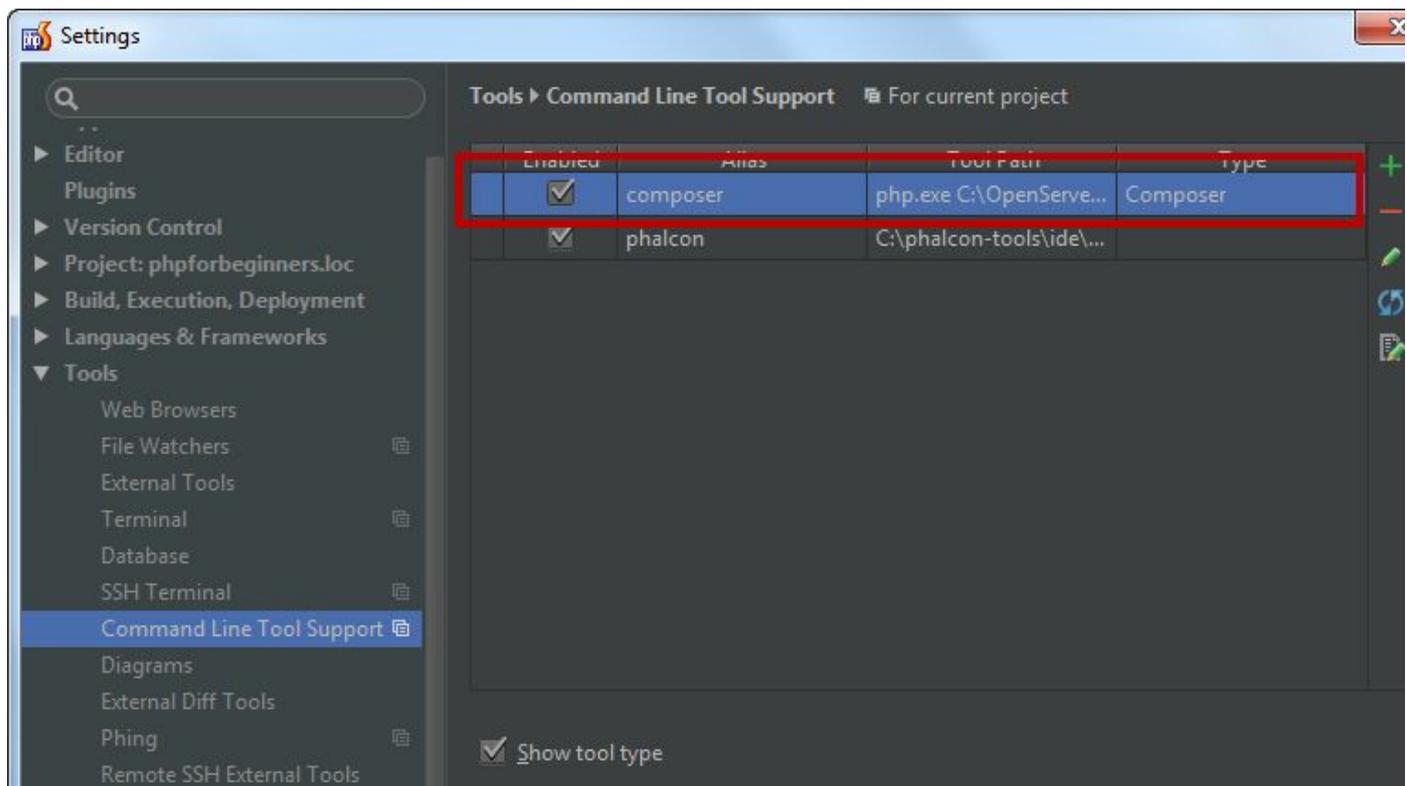
В открывшемся окне выбираем из списка Composer, в параметре Visibility - выбираем project



В новом окне выбираем версию PHP а также указываем папку, в которой находится файл composer.phar и нажимаем OK.

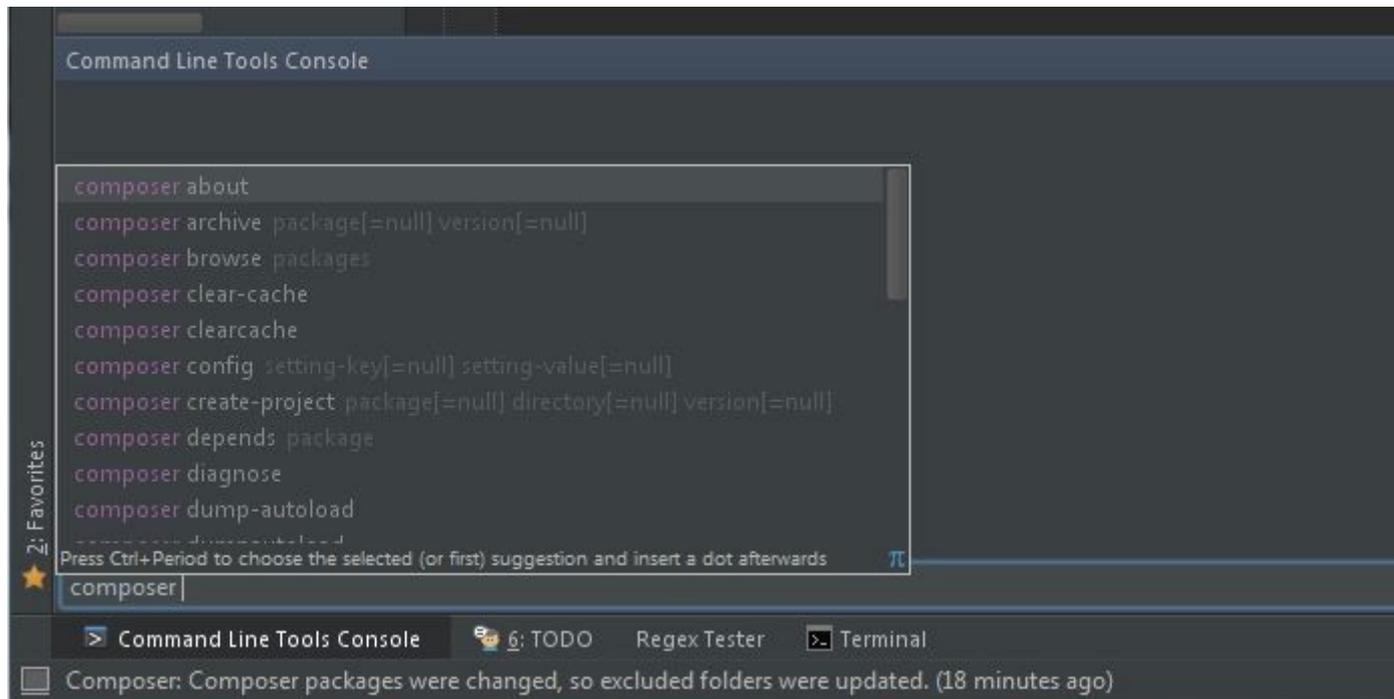


У нас появится запись о том, что теперь в нашем проекте есть поддержка composer и его можно использовать с псевдонимом (алиасом) с. Имя этого алиасом можно также отредактировать, нажав на кнопку отредактировать.



В качестве алиаса установим значение composer и нажмем OK.

После, нажимаем сочетание клавиш [CNTRL] + [SHIFT] + X. В открывшейся вкладке Command Line Tools console теперь после ввода название алиаса composer PhpStorm показывает автодополнение ко всем командам, которые поддерживает composer.



Сочетания клавиш

[CTRL] + [SHIFT] + [A] - данная команда вызывает окно, с помощью которого можно найти любую настройку в самой IDE PhpStorm

[CTRL] + [F] — поиск в текущем файле

[CTRL] + [SHIFT] + [F] — поиск по файлам (можно указать в каких папках искать)

[CTRL] + [N] — поиск по имени класса

[CTRL] + [SHIFT] + [N] — поиск по имени файла

[CTRL] + [SHIFT] + [ALT] + [N] — поиск любой функции/метода и констант (полезно, если помните название метода, но не помните класс, в котором он записан)

[CTRL] + [O] - данная команда вызывает окно с помощью которого можно выбрать, какой метод нужно переопределить в данном классе

[CTRL] + [I] - данная команда вызывает окно с помощью которого можно выбрать, какой “магический” метод нужно переопределить в данном классе

[CTRL]+ [ALT] + [Space] — показать подсказку

[CTRL]+ [ALT] + [T] — окружить выделенный код блоком

[ALT] + [ENTER] — окружить выделенный код блоком

[CTRL] + / — комментировать / раскомментировать (!) выделенные строчки (или текущую если нет выделения) ([CTRL] + [SHIFT] + / — то же, но блочным комментарием), при этом обращаем внимание, что команда не работает в русском регистре, в этом случае она сворачивает блок.

[CTRL] + [ALT] + [V]— вынести выделенный блок кода в переменную (и заменить на использование здесь и всюду)

[CTRL] + [Q]— Выделите/поставьте курсор на метод/переменную/константу/функцию/класс и после комбинации отобразится документация по этому элементу.

Табы / Вкладки / навигация [ALT] + [LEFT/RIGHT] — Быстрое переключение таба на левый или правый. [CTRL] + [TAB] / [CTRL] + [SHIFT]+[TAB] — Переход к открытым табам и внутренним окнам. (Как и во многих приложениях)

[CTRL] + [ALT] + [LEFT/RIGHT] - Переход по истории где вы побывали когда перемещались по файлам назад/вперед. При этом курсор возвращается к последнему месту редактирования в файле.

[CTRL] + [B] — Перейти к месту определения сущности указанной в коде (метод, переменная, и т.д.). Тоже самое, что и [CTRL]+ клик левой кнопкой мышкой.

[CTRL] + [SHIFT] +[V] — Вставить из истории буфера обмена.

[SHIFT] +[F6] — Переименование выделенного метода

Дополнительно настраиваем (не стандартно, но так привыкли) : в пару к CTRL+D дублировать строку (блок) CTRL+E — Удалить строку

[SHIFT] +[SPACE] - Вызов автодополнения

[CTRL] + [SHIFT] + [ENTER] - Вставляет выбранный элемент (класс/метод /переменная и т.д.) и вставляет в текущую строку с добавлением ";"

[CTRL] + [P] - Показывает информацию о аргументах функции / метода

[CTRL] + [F1] - Просмотр описания ошибки в текущей строке

[ALT] + [INSERT] - Вызов контекстного меню, которое позволяет сгенерировать множество полезных вещей

[CTRL] + [I] - Инструмент аналогичный предыдущему для определения методов

[CTRL] + [/] - Закомментировать/Раскомментировать выделенный фрагмент кода с помощью //

[CTRL] + [SHIFT] + [/] - Аналогично предыдущему, но используется блочный комментарий /**/

[CTRL] + [W] - Выделение содержимого текущего блока

[CTRL] + [SHIFT] + [W] - Возврат текущего выделения в предыдущее состояние

[CTRL] + [ALT] + [L] - Бьютифер кода, особенно полезен, когда нужно разбираться в коде низкого качества и ужасного форматирования.

[CTRL] + [D] - Дубликат текущей строки

[CTRL] + [Y] - Удаление текущей строки

[CTRL] + [SHIFT] + [J] - Удаление лишних переносов строк

Работа с GIT

[CTRL] + [K] - коммит в репозиторий

[CTRL] + [SHIFT] + [K] - push в удаленный репозиторий

Метки (полезно пометить какую-то строку в файле, чтобы потом к ней перейти)

[CTRL]+ [SHIFT] + (0...9) — Установить/сбросить метку в проекте

[CTRL]+ (0...9) — перейти к метке

[F11] - установка (удаление) закладки в коде файла

[CTRL]+ [F11] - перемещение к закладке в коде

TODO (в коде в комментариях удобно писать //todo...)

[ALT] + [6] — открыть окно TODO

[CTRL] + [ALT] + [Up][Down] — перемещаться по TODO