

jenyay.net

Софт, исходники и фото

Поиск:

>>

[Домой](#) [Блог](#) [Контакты](#)[Печать](#) [Править](#)

Блог

Программки

OutWiker (rus)

[Плагины](#)[Бета-версии](#)[Локализации](#)[Документация](#)[Предложения и](#)[баги](#)[Исходники](#)

OutWiker (en)

[Plug-ins](#)[Beta versions](#)[Translate](#)[Suggestions and](#)[bugs](#)[Source code](#)[Documentation](#)

Другие...

Программирование

[Python](#)[Rust](#)[.NET/C#](#)[C++](#)[PHP](#)[Алгоритмы](#)[Инструменты](#)[Остальное](#)

Обзоры книг

Программирование скриптов для Vim. Часть 1. Запуск скриптов

- [Введение](#)
- [Командная строка Vim](#)
- [Выполнение открытого в данный момент скрипта](#)
- [Выполнение скриптов внутри файлов .vimrc/ vimrc](#)
- [Специальные папки](#)
- [Комментарии](#)

Введение

Vim замечателен тем, что к нему создано огромное количество плагинов и различных скриптов-примочек, но когда в руки попадает такой мощный инструмент, то всегда хочется сделать что-то такое, что еще больше облегчило бы его использование. Эта статья, которая будет состоять из нескольких частей, должна поможет научиться писать свои скрипты для Vim/gVim.

Vim имеет свой собственный язык для скриптов, кроме того в этом редакторе можно использовать язык Python. Python мы в первое время трогать не будем, а займемся изучением встроенного языка Vim. Прежде всего рассмотрим способы запуска скриптов.

Командная строка Vim

Самый простой способ запуска - это воспользоваться командной строкой Vim. В обычном режиме нажимаем ":", чтобы попасть в командную строку и вводим там команду

Студентам

Фото

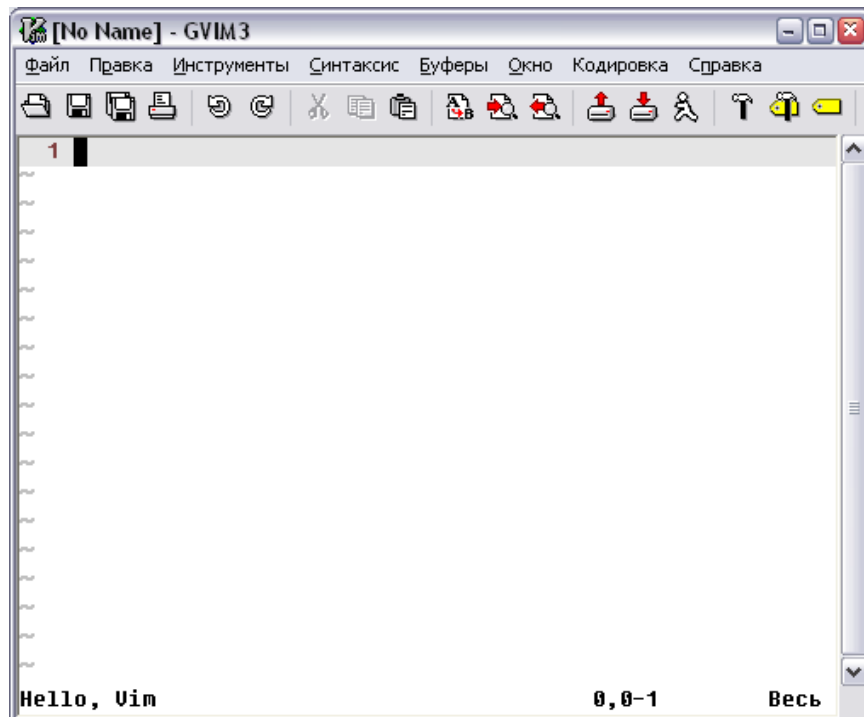
Животные
Черно-белые
Пейзажи/Природа
Город
Закаты
Панорамы
Спорт
Репортаж
Разное

Контакты

```
:echo "Hello, Vim"
```

[Исходник](#)

После нажатия Enter в той же командной строке мы увидим:



Как вы уже, наверное, догадались, команда *echo* как раз и выполняет отображение строк. Заметьте, что эта команда выводит текст в командную строку (в случае выполнения *echo* из *.vimrc* в виде диалогового сообщения), а не в открытый текстовый файл. Для добавления текста в файл используется другая команда *append*, но мы ее пока касаться не будем.

В командной строке можно также запускать скрипты, состоящие из многих строк, вводя по одной строке. Но такой способ удобен, в основном, для ввода коротких команд. А в большинстве случаев удобнее сохранить скрипт в файл и уже его запускать. Давайте посмотрим как это делается.

Для запуска скриптов, сохраненных в файл есть встроенная команда *source* или в сокращенном варианте *so*. В дальнейшем для наглядности мы везде будем использовать полные варианты.

Теперь создадим текстовый файл с тем же содержанием и назовем его *hello.vim*.

```
echo "Hello, Vim"
```

[Исходник](#)

Обратите внимание, что здесь уже нет символа ":" перед командой *echo*.

Теперь мы можем в командной строке Vim выполнить следующую команду:

```
:source полный_путь_до_скрипта\hello.vim
```

[Исходник](#)

В результате увидим ту же картину, что и на предыдущем скриншоте.

Здесь хотелось бы немного отвлечься и поговорить о путях к скриптам. Чтобы каждый раз не вводить полный путь до папки, где у нас будут храниться скрипты, их можно расположить в одной из специальных папок, путь до которых Vim уже знает. Переменные, хранящие такие пути начинаются с символа "\$" и состоят из заглавных букв. Правда, эти переменные можно писать и маленькими буквами, но лучше придерживаться написания, принятом в документации к Vim. Чтобы не было недоразумения, сразу скажу, что в именах обычных переменных регистр символов имеет важное значение и переменные *A* и *a* - это разные переменные.

Рассмотрим в качестве примера некоторые из таких переменных, хранящих пути:

Переменная	Описание	Пример пути в Windows
\$VIM	Путь до установленного Vim	C:\Program Files\Vim
\$VIMRUNTIME	Путь до исполняемого файла Vim	C:\Program Files\Vim\vim72
\$HOME	Путь до домашней папки	C:\Documents and Settings\USERNAME

Значения этих переменных можно изменять в процессе работы, но не думаю, что это будет очень уж удачное решение.

Итак, где же нам хранить наши скрипты? Если мы хотим, чтобы доступ к ним имели все пользователи компьютера, то можно создать папку *scripts*, например, внутри папки *\$VIM* или *\$VIMRUNTIME*, в этом случае для запуска скрипта *hello.vim* достаточно будет ввести команду:

```
:source $VIM\scripts\hello.vim
```

[Исходник](#)

или

```
:source $VIMRUNTIME\scripts\hello.vim
```

[Исходник](#)

Если же мы хотим, чтобы доступ к скрипту имел только текущий пользователь, или у нас нет прав для записи в *\$VIM* или *\$VIMRUNTIME*, то можно создать папку *scripts* в домашней директории.

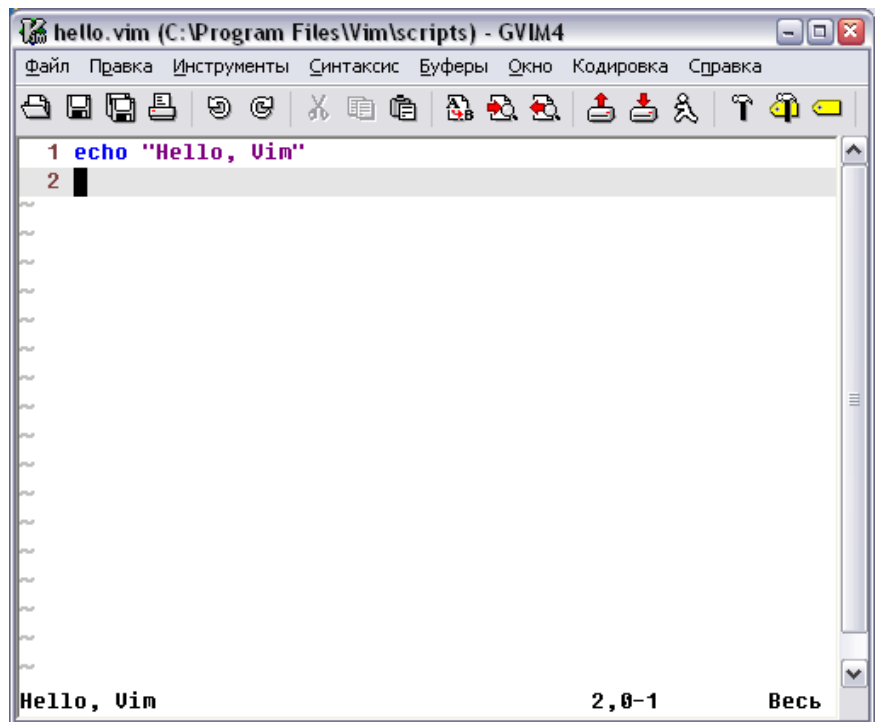
Выполнение открытого в данный момент скрипта

Рассмотрим еще один вариант запуска скриптов, который может быть удобен при его написании и отладке. Не смотря на то, что запуск все-равно происходит через командную строку, этот способ хотелось бы выделить отдельно. Vim позволяет выполнять скрипты, открытые в текущем буфере. Для этого откроем наш файл *hello.vim* в самом редакторе Vim и выполним следующую команду через командную строку:

```
:source %
```

[Исходник](#)

В результате наш скрипт запустится и мы увидим ожидаемую надпись:



Выполнение скриптов внутри файлов `.vimrc/_vimrc`

Вы, наверное, уже обратили внимание, что при нам уже приходилось писать маленькие скрипты. Записи, сделанные в файле `.vimrc` (или `_vimrc` в Windows) и есть те самые команды, которые выполняет `vim`, но запускаются они при каждом запуске редактора. Часто именно это нам и нужно, но бывают ситуации, когда скрипты нужно запускать во время работы, возможно и не один раз. Поэтому давайте рассмотрим другие способы запуска скриптов.

Посмотрим как будет работать та же команда `echo` внутри файла `.vimrc/_vimrc`. Откроем для редактирования файл настроек, это можно сделать с помощью команды

```
:edit $MYVIMRC
```

[Исходник](#)

После этого добавим в конец ту же самую строку

```
echo "Hello, Vim"
```

[Исходник](#)

Теперь при запуске `gVim` мы будем видеть такое вот окошко с сообщением:



Но для большего интереса мы можем запускать из `.vimrc/_vimrc` наш скрипт `hello.vim`. Например, таким образом (в данном случае скрипт у меня лежит в папке `$VIM/scripts`)

```
source $VIM/scripts/hello.vim
```

[Исходник](#)

В результате при запуске gVim будет открываться такое же окошко с сообщением "Hello, Vim". Еще обратите внимание, что в путях я использовал в данном случае не обратный, а прямой слеш "/". В Windows-версии Vim слеш можно использовать как прямые, так и обратные.

Специальные папки

В реальности для автозапуска скриптов при запуске Vim лучше воспользоваться специальной папкой `plugin`, все скрипты из которой выполняются при запуске Vim. Вы можете удалить упоминание `hello.vim` в вашем файле `.vimrc/_vimrc` и просто скопировать файл `hello.vim` в папку `plugins`, результат будет тот же самый.

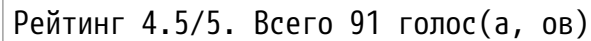
Также есть папка `ftplugin`, в которой расположены скрипты, выполняющиеся только для открытых файлов определенного типа (сpp, py, html и т.д.), но мы их пока касаться не будем.

Подведем некоторые итоги и еще раз кратко перечислим возможные способы выполнения скриптов в Vim

- Выполнение скриптов построчно через командную строку Vim.
- Выполнение скриптов, сохраненных в отдельных файлах с помощью команды `source`.
- Выполнение текущего открытого скрипта с помощью команды `source %`.
- Выполнение скрипта из файла `.vimrc/_vimrc`.

- На этом мы завершим первую часть статьи. Продолжение следует, в следующей главе мы будем учиться работать с переменными и познакомимся с существующими в Vim областями видимости переменных.

Вы можете подписаться на новости сайта через [RSS](#), [Группу Вконтакте](#) или [Канал в Telegram](#).



☐ Плохо ☐ Так себе ☐ Неплохо ☐ Хорошо ☐ Отлично

[Подписаться на комментарии](#)

Автор:

Тема:

Ваш комментарий

[illegible]

Введите код

Послать