Konfiguracja i ściągawka

Plik .vimrc i ściągawka do wydruku.

- Konfiguracja: plik .vimrc
- Ściągawka do wydruku

Konfiguracja: plik .vimrc

Poziom: średni **Dotyczy: Vim 9.x** **Czas: ~20 min**

Ustawienia wpisywane komendą `:set` znikają, gdy zamkniemy Vima. Plik `.vimrc` zapisuje je raz na zawsze, więc każda kolejna sesja startuje już z numeracją linii, podświetleniem składni i resztą ustawień gotowych do pracy.

Skróty użyte na tej stronie (1)

Skrót	Co znaczy
YAML	YAML Ain't Markup Language: format plików konfiguracyjnych oparty na wcięciach

Fiszka

TYLKO ODCZYT

Czym jest plik konfiguracyjny edytora

```
~/.vimrc
```

> **Pojęcie: plik konfiguracyjny.** Zwykły plik tekstowy, który program czyta przy starcie i stosuje zapisane w nim ustawienia, zamiast wymagać ich ręcznego wpisania za każdym razem. `.vimrc` to plik konfiguracyjny Vima: każda linia w nim to komenda, którą inaczej wpisalibyśmy ręcznie po dwukropku.

TYLKO ODCZYT

Lokalizacja dla zwykłego użytkownika: ~/.vimrc

```
vim ~/.vimrc
```

Plik w katalogu domowym konta, na przykład `/home/nazwa_usera/.vimrc`. Ustawienia w nim obowiązują tylko to jedno konto, na tym jednym serwerze.

vim = Vi IMproved: ulepszony edytor vi

TYLKO ODCZYT

Lokalizacja dla roota i dla całego systemu: /etc/vimrc

```
sudo vim /etc/vimrc
```

`/etc/vimrc` (albo `/etc/virc` w niektórych dystrybucjach) czytają wszyscy użytkownicy systemu, także root, jeśli nie mają własnego `~/.vimrc`. Własny plik konta ma pierwszeństwo: jego ustawienia dokładają się do systemowych i mogą je nadpisać.

vim = Vi IMproved: ulepszony edytor vi

ZMIENIA SYSTEM

Gotowy .vimrc dla administratora

```
cat > ~/.vimrc <<'EOF'
" Numery linii z lewej strony ekranu
set number

" Podswietlanie skladni odpowiednie do typu pliku
syntax on

" Wyszukiwanie przyrostowe: kursor skacze do dopasowania
" juz w trakcie wpisywania wzorca, bez czekania na Enter
set incsearch

" Wciecia spacjami zamiast tabulatora, wazne w YAML,
" gdzie tabulator w ogole nie jest dozwolony w skladni
set expandtab
set shiftwidth=2
set tabstop=2

" Widoczne tabulatory i spacje na koncu linii,
" pomaga zlapac literowki niewidoczne na pierwszy rzut oka
```

```
set list
set listchars=tab:>-,trail:.

" Dlugie linie nie zawijaja sie na granicy okna
set nowrap
EOF
```

Plik gotowy do skopiowania na nowy serwer, komentarze zaczynające się od `"` Vim ignoruje przy wczytywaniu.

cat = concatenate: sklej i wypisz pliki · **YAML** = YAML Ain't Markup Language: format plików konfiguracyjnych oparty na wcięciach

ZMIENIA SYSTEM

Ustawienie na chwilę: :set w bieżącej sesji

```
:set number
```

Wpisane w trybie wiersza poleceń działa tylko do zamknięcia edytora. Wygodne, gdy chcemy sprawdzić jedno ustawienie bez zmieniania stałej konfiguracji.

ZMIENIA SYSTEM

Ustawienie na stałe: wpis w .vimrc

```
set number
```

Ta sama komenda, bez początkowego dwukropka, wpisana jako osobna linia w `.vimrc`, obowiązuje od tej pory w każdej nowej sesji Vima na tym koncie.

ZMIENIA SYSTEM

Tryb wklejania: set paste

```
:set paste
```

Włączamy zawsze przed wklejeniem czegokolwiek do Vima przez schowek terminala, nigdy po. Bez tego ustawienia Vim traktuje każdy wklejany znak nowej linii tak, jakby użytkownik

naciskał Enter ręcznie, i dokłada do każdej kolejnej linii automatyczną autoindentację odziedziczoną z linii poprzedniej.

Przykład błędnie wklejonego YAML bez `set paste`, plik już nie parsuje się poprawnie:

```
services:
  web:
      image: nginx
    db:
        image: postgres
```

Ten sam fragment wklejony po `:set paste`:

```
services:
  web:
    image: nginx
  db:
    image: postgres
```

Każda kolejna linia dostawała dodatkowe wcięcie, bo autoindentacja doliczała wcięcie poprzedniej linii do każdej nowej. W YAML, gdzie poziom wcięcia jest częścią składni, tak rozjechany plik przestaje się poprawnie parsować, a komunikat błędu z narzędzia, które go czyta później, zwykle nie mówi wprost, że problemem jest wcięcie.

> Po wklejeniu warto wrócić do `:set nopaste`, żeby zwykłe pisanie tekstu znowu korzystało z autoindentacji tam, gdzie jest pomocna.

YAML = YAML Ain't Markup Language: format plików konfiguracyjnych oparty na wcięciach

Z życia wzięte

Z ŻYCIA WZIĘTE

Tomek dostaje od Oli gotowy fragment `compose.yaml` przez czat i wkleja go do otwartego już w Vimie pliku na `docker01`. Po zapisaniu Docker odmawia uruchomienia kontenera z błędem `mapping values are not allowed here`. Dopiero `vimdiff` z oryginałem pokazuje, że każda kolejna linia wklejonego fragmentu ma jeden dodatkowy poziom wcięcia więcej niż poprzednia, bo Tomek wkleił tekst bez wcześniejszego `:set paste`.

Jak to działa

> **Pojęcie: sesja i plik konfiguracyjny.** Ustawienia zmienione komendą `:set` żyją tylko w pamięci bieżącej sesji Vima i znikają razem z nią. `.vimrc` jest czytany raz, przy starcie każdej nowej sesji, więc zapisane w nim ustawienia działają zawsze od nowa, bez potrzeby wpisywania ich ręcznie. To ten sam mechanizm, co plik `~/.bashrc` dla powłoki: różnica jest tylko w tym, który program go czyta.

Częste błędy

Objaw	Przyczyna	Co robimy
Wklejony YAML przestaje się parsować po wklejeniu do Vima	brak `:set paste` przed wklejeniem, autoindentacja rozjechała wcięcia	`:set paste` przed kolejną próbą, albo `:e!`, żeby wczytać plik od nowa, jeśli jeszcze nie zapisaliśmy
Ustawienie zniknęło po ponownym otwarciu Vima	wpisane komendą `:set` w bieżącej sesji, nie zapisane w `.vimrc`	dopisujemy tę samą linię, bez dwukropka, do `~/.vimrc`
`.vimrc` nie działa mimo poprawnych ustawień w środku	plik leży w złym katalogu albo konto nie ma do niego dostępu	sprawdzamy `ls -la ~/.vimrc` i uprawnienia do odczytu dla właściciela pliku

□□ Ćwiczenia

1. Chcemy, żeby numeracja linii i podświetlanie składni włączały się automatycznie w każdej nowej sesji Vima na naszym koncie. Gdzie i co zapisujemy?

Odpowiedź

W pliku `~/.vimrc` dopisujemy dwie linie: `set number` oraz `syntax on`, bez początkowego dwukropka. Plik jest czytany przy każdym starcie Vima na tym koncie.

1. Dlaczego wklejenie sformatowanego bloku YAML do Vima bez wcześniejszego `:set paste` może zepsuć plik, mimo że sam tekst źródłowy był poprawny?

Odpowiedź

Bez `:set paste` Vim traktuje każdy znak nowej linii we wklejanym tekście tak, jakby użytkownik nacisnął Enter ręcznie, i dokłada do każdej nowej linii automatyczne wcięcie wynikające z poprzedniej. Kolejne linie wklejanego bloku dostają narastające, dodatkowe wcięcia, mimo że w źródle miały wcięcia równe, a w YAML poziom wcięcia jest częścią składni.

Powiązane: Pierwsze kroki: otwarcie, zapis, wyjście · Wiele plików: bufory, okna, karty · Pliki konfiguracyjne i formaty: YAML, INI, systemd unit

Źródła: `man 1 vim` · `:help vimrc.txt`, `:help usr_05.txt` w samym Vimie

Ściągawka do wydruku

Poziom: podstawy **Dotyczy: Vim 9.x** **Czas: ~5 min**

Jedna strona ze wszystkimi skrótami z tej książki, pogrupowana w tabele do szybkiego zerknięcia. Do wydrukowania i powieszenia przy monitorze, bez scenek i bez tłumaczeń, które są na pozostałych stronach książki.

Wejście i wyjście

Klawisz albo komenda	Działanie
`vim SciezkaDoPliku`	otwarcie pliku
`i`, `a`, `o`, `O`	wejście w tryb wstawiania (przed, za, linia niżej, linia wyżej)
`Esc`	powrót do trybu normalnego
`:w`	zapis bez wyjścia
`:wq`, `:x`, `ZZ`	zapis i wyjście
`:q!`	wyjście bez zapisu, jedno okno
`:qa!`	wyjście bez zapisu, wszystkie okna
`:w !sudo tee % > /dev/null`	zapis pliku otwartego bez sudo

Tryby

Klawisz	Wejście w tryb
`i` `a` `o` `O`	wstawiania
`v` `V` `Ctrl+v`	wizualny (znakowy, liniowy, blokowy)
`:`	wiersza poleceń
`R`	zastępowania
`Esc`	powrót do normalnego z dowolnego innego

Poruszanie się

Klawisz	Ruch
`h` `j` `k` `l`	lewo, dół, góra, prawo
`w` `b` `e`	początek następnego słowa, początek poprzedniego, koniec bieżącego
`0` `^` `$`	początek linii, pierwszy niepusty znak, koniec linii
`gg` `G`	początek pliku, koniec pliku
`:N`, `NG`	skok do linii N
`Ctrl+f` `Ctrl+b`	strona w przód, strona w tył
`%`	pasujący nawias
`ma`, `'a`	zapisanie i powrót do znacznika `a`

Edycja

Klawisz albo komenda	Działanie
`x`	kasowanie znaku
`dd`, `3dd`	kasowanie linii (z licznikiem: N linii)
`yy`	kopiowanie linii
`p` `P`	wklejenie po, wklejenie przed
`u` `Ctrl+r`	cofnięcie, ponowienie
`.`	powtórzenie ostatniej zmiany
`dw` `d$` `c$` `dG`	operator plus ruch: usuń słowo, usuń do końca linii, zmień do końca linii, usuń do końca pliku
`>>` `<<`	zwiększenie, zmniejszenie wcięcia

Szukanie i zamiana

Komenda	Działanie
`/wzorzec`, `?wzorzec`	szukanie w dół, szukanie w górę

Komenda	Działanie
`n` `N`	kolejne dopasowanie, w tym samym kierunku i w przeciwnym
`*` `#`	szukanie słowa pod kursorem w dół, w górę
`:%s/a/b/g`	zamiana we wszystkich wystąpieniach w całym pliku
`:%s/a/b/gc`	to samo, z potwierdzeniem każdego dopasowania
`:N,Ms/a/b/g`	zamiana ograniczona do linii od N do M

Wiele plików

Komenda	Działanie
`:e SciezkaDoPliku`	otwarcie kolejnego pliku
`:ls`	lista otwartych buforów
`:bn` `:bp`	następny bufor, poprzedni bufor
`:sp` `:vsp`	podział okna poziomo, pionowo
`Ctrl+w Ctrl+w`	przełączenie między oknami
`:tabnew SciezkaDoPliku`	nowa karta
`vimdiff PlikA PlikB`	porównanie dwóch plików

Makra, rejestry, bloki

Komenda	Działanie
`qa ... q`	nagranie makra do rejestru `a`
`@a`, `5@a`	odtworzenie makra, pięć razy z rzędu
`@@`	powtórzenie ostatnio odtworzonego makra
`"ayy`, `"ap`	kopiowanie i wklejanie z rejestru nazwanego `a`
`Ctrl+v`, ruch, `I`, tekst, `Esc`	edycja blokowa wielu linii naraz
`:g/wzorzec/d`	wykonanie komendy na każdej pasującej linii
`:sort`	sortowanie linii pliku
`:!komenda`	uruchomienie komendy powłoki bez wychodzenia

Konfiguracja

Komenda	Działanie
`~/.vimrc`	plik konfiguracyjny dla jednego konta
`/etc/vimrc`	plik konfiguracyjny dla całego systemu
`set number`	numery linii
`syntax on`	podświetlanie składni
`set expandtab`, `set shiftwidth=2`	wcięcia spacjami, ważne w YAML
`:set paste` przed wklejaniem, `:set nopaste` po	bez rozjeżdżania wcięć wklejonego tekstu

Powiązane: Vim: edytor, z którego nie da się wyjść · Pierwsze kroki: otwarcie, zapis, wyjście