Pierwsze kroki: otwarcie, zapis, wyjście

Poziom: podstawy **Dotyczy: Vim 9.x** **Czas: ~15 min**

To jest najważniejsza strona całej książki. Zanim nauczymy się czegokolwiek więcej, musimy umieć bezpiecznie otworzyć plik, zapisać zmiany i wyjść, bez zamykania całego okna terminala jak ostatniej deski ratunku.

Fiszka

TYLKO ODCZYT

Otwarcie pliku: vim

```
vim /etc/fstab
```

`vim SciezkaDoPliku` otwiera plik w trybie normalnym. W tym trybie klawisze nie wpisują tekstu, tylko sterują edytorem, więc samo otwarcie pliku niczego nie zmienia, nawet gdy naciśniemy kilka niewłaściwych klawiszy.

Przykład:

```
"/etc/fstab" 12L, 512B
```

vim = Vi IMproved: ulepszony edytor vi

TYLKO ODCZYT

Wejście w tryb wstawiania: i

```
i
```

`i` przełącza z trybu normalnego w tryb wstawiania dokładnie w miejscu kursora, dolny pasek pokazuje wtedy `-- INSERT --`. Warianty `a` (wstawiamy za kursorem), `o` (otwiera nową,

pustą linię poniżej) i `O` (otwiera linię powyżej) robią to samo, tylko zaczynają pisanie w innym miejscu.

□□ TYLKO ODCZYT

Wyjście z trybu wstawiania: Esc

```
Esc
```

<kbd>Esc</kbd> zawsze i bezwarunkowo wraca do trybu normalnego, niezależnie od tego, w jakim trybie akurat się znajdujemy. To jest pierwszy odruch w każdej sytuacji, w której nie wiadomo, co się dzieje na ekranie.

□□ TYLKO ODCZYT

Zapis bez wychodzenia: :w

```
:w
```

`:w` zapisuje bufor na dysk i zostawia nas w edytorze, gotowych do dalszej pracy. Wpisujemy to poleceniami z dwukropkiem z trybu normalnego, nigdy z trybu wstawiania.

□□ TYLKO ODCZYT

Zapis i wyjście: :wq, :x, ZZ

```
:wq
:x
ZZ
```

Wszystkie trzy zapisują plik i zamykają edytor, ale nie identycznie. `:wq` zapisuje zawsze, nawet gdy nic nie zmieniliśmy. `:x` i `ZZ` (dwa wielkie Z naciśnięte w trybie normalnym, bez dwukropka) zapisują tylko wtedy, gdy plik faktycznie się zmienił, co jest odrobinę szybsze i nie odświeża niepotrzebnie znacznika czasu pliku.

□□ TYLKO ODCZYT

Wyjście bez zapisywania: :q!

```
:q!
```

`:q!` wymusza wyjście z porzuceniem wszystkich niezapisanych zmian, bez żadnego pytania. To jest komenda, której szukamy, gdy coś w pliku poszło nie tak i wolimy zacząć od nowa niż ratować bieżący stan.

TYLKO ODCZYT

Wyjście ze wszystkich okien: :qa!

```
:qa!
```

`:qa!` różni się od `:q!` zakresem: `:q!` zamyka jedno, bieżące okno albo kartę, `:qa!` (quit all) zamyka naraz wszystkie otwarte okna i karty tej samej sesji Vima, porzucając zmiany we wszystkich z nich. Gdy pracujemy tylko na jednym pliku, oba polecenia dają ten sam efekt, różnica wychodzi na jaw dopiero po otwarciu kilku plików jednocześnie, co pokazuje strona Wiele plików: bufory, okna, karty.

ZMIENIA SYSTEM

Zapis pliku otwartego bez sudo: :w !sudo tee %

```
:w !sudo tee % > /dev/null
```

Ratunek, gdy otworzymy plik systemowy zwykłym `vim plik` bez `sudo`, wprowadzimy zmiany, i dopiero przy `:w` dostaniemy błąd braku uprawnień. `%` to symbol bieżącego pliku, cała komenda przepuszcza zawartość bufora przez `sudo tee`, zapisując ją z podwyższonymi uprawnieniami, bez zamykania edytora i bez utraty pracy.

Przykład:

```
E212: Can't open file for writing
```

TYLKO ODCZYT

Interaktywny samouczek: vimtutor

```
vimtutor
```

Osobny program instalowany razem z pełnym pakietem Vima, kopiuje plik ćwiczeniowy do katalogu tymczasowego i prowadzi przez piętnaście do dwudziestu minut praktycznych ćwiczeń z wejściem, zapisem i wyjściem opisanymi wyżej. Najszybszy sposób, żeby powyższe komendy faktycznie weszły w palce, zamiast zostać wiedzą teoretyczną.

vimtutor = samouczek Vima: interaktywny program instalowany razem z Vimem, uczy podstaw w piętnaście do dwudziestu minut

Z życia wzięte

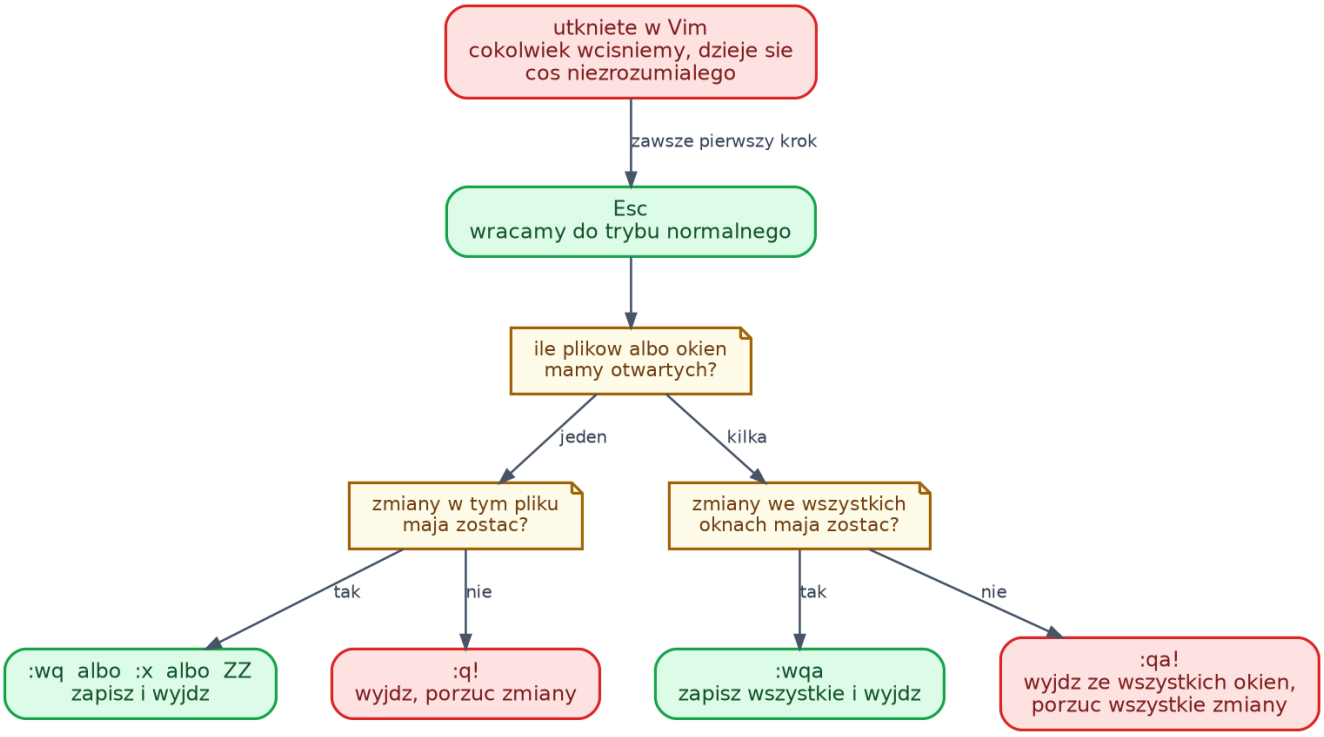
Z ŻYCIA WZIĘTE

Drugi tydzień pracy Tomka. Dostaje zadanie poprawienia wpisu w `/etc/fstab` na `docker01`, żeby dysk z danymi montował się pod właściwą nazwą. Wpisuje `sudo vim /etc/fstab`, poprawia literówkę klawiszem `i`, a potem przez piętnaście minut próbuje wyjść, naciskając na przemian <kbd>Ctrl</kbd>+<kbd>C</kbd> i samo <kbd>Esc</kbd>, aż w końcu zamyka całe okno terminala razem z niezapisaną poprawką.

Jak to działa

Pojęcie: tryb. Vim jest edytorem modalnym: te same klawisze robią różne rzeczy w zależności od tego, w jakim trybie akurat pracujemy. W trybie normalnym litera `i` przełącza tryb, w trybie wstawiania ta sama litera po prostu wpisuje znak „i" do tekstu. Pełne omówienie wszystkich trybów i tego, jak poznać, w którym jesteśmy, czeka na następnej stronie: Tryby: dlaczego klawisze robią różne rzeczy.

Gdy zupełnie nie wiadomo, co się dzieje, poniższy schemat prowadzi krok po kroku do bezpiecznego wyjścia z dowolnej sytuacji.



Jak wyjść z Vima: drzewo decyzyjne od utknięcia do zapisu albo porzucenia zmian (kliknięcie otwiera pełny rozmiar)

Częste błędy

Objaw	Przyczyna	Co robimy
Nie wiemy, w jakim jesteśmy trybie, klawisze robią coś dziwnego	zgubiliśmy się między trybem normalnym a wstawiania	<kbd>Esc</kbd>, sprawdzenie dolnego paska (`-- INSERT --` oznacza tryb wstawiania, jego brak, tryb normalny)
`:q` odmawia wyjścia komunikatem `E37: No write since last change`	są niezapisane zmiany, Vim celowo nie pozwala ich stracić przez przypadek	`:wq`, jeśli zmiany mają zostać, `:q!`, jeśli mają zostać porzucone
Wpisanie `qa!` w trybie normalnym nic nie robi albo zaczyna nagrywać makro	`qa!` bez dwukropka to nie jest polecenie wyjścia, w trybie normalnym `q` zaczyna nagrywanie makra do rejestru, a litera po nim wskazuje ten rejestr	poprawna komenda to `:q!`, z dwukropkiem na początku
`:w` kończy się `E212: Can't open file for writing`	plik systemowy otwarty bez `sudo`	`:w !sudo tee % > /dev/null`, bez zamykania edytora

Ćwiczenia

1. Otwieramy dowolny plik tekstowy przez `vim`, dopisujemy jedną linię i chcemy ją zapisać, zostając w edytorze do dalszej pracy. Jakiej komendy używamy?

Odpowiedź

`:w`. Zapisuje bufor na dysk bez zamykania edytora, w przeciwieństwie do `:wq`, które od razu kończy sesję.

1. Wprowadziliśmy zmiany w pliku, uznaliśmy, że są błędne, i chcemy wrócić do wersji sprzed otwarcia edytora, bez zapisywania niczego. Jaka jest właściwa komenda i dlaczego stara notatka `qa!` nie zadziała?

Odpowiedź

Właściwa komenda to `:q!`. `qa!` bez dwukropka nie jest poleceniem trybu wiersza poleceń: w trybie normalnym Vim odczyta literę `q` jako początek nagrywania makra, `a` jako nazwę rejestru, do którego to makro ma trafić, a `!` jako zwykły znak do wpisania przy najbliższym wejściu w tryb wstawiania. Nic z tego nie zamyka edytora.

Powiązane: Tryby: dlaczego klawisze robią różne rzeczy · Wiele plików: bufory, okna, karty · Uprawnienia do plików i sudo · Konfiguracja: plik .vimrc · Ściągawka do wydruku

Źródła: `man 1 vim` · `:help quitting` w samym Vimie

Revision #32
Created 2025-03-07 13:06:07 UTC by Tomasz Konieczny
Updated 2026-10-26 14:07:21 UTC by Tomasz Konieczny