Edycja: kasowanie, kopiowanie, wklejanie, cofanie

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

Vim nie ma osobnego klawisza na każdą możliwą operację edycji. Zamiast tego łączy operator z ruchem, a te same kilka liter daje kombinatorycznie ogromny zestaw komend. Ta strona pokazuje tę gramatykę i najczęściej używane skróty, które z niej korzystają.

Fiszka

TYLKO ODCZYT

Kasowanie znaku: x

```
x
```

Kasuje jeden znak pod kursorem. Najprostsza operacja edycji, bez operatora i ruchu, przydatna do poprawiania literówek.

TYLKO ODCZYT

Kasowanie linii: dd

```
dd
3dd
```

`dd` kasuje całą bieżącą linię. Licznik przed komendą powtarza ją tyle razy, ile wskazuje liczba: `3dd` kasuje trzy linie naraz, licząc od bieżącej w dół.

TYLKO ODCZYT

Kopiowanie linii: yy

```
yy
```

`yy` (yank) kopiuje bieżącą linię do rejestru, bez jej kasowania. Działa razem z `p` i `P` poniżej.

TYLKO ODCZYT

Wklejanie: p P

```
p
P
```

`p` wkleja zawartość rejestru po bieżącej linii (albo po kursorze, przy wklejaniu w środku linii), `P` przed nią. Rejestr wypełnia poprzednie `dd` albo `yy`.

TYLKO ODCZYT

Cofanie i ponawianie: u Ctrl+r

```
u
Ctrl+r
```

`u` (undo) cofa ostatnią zmianę, można nacisnąć wielokrotnie, żeby cofnąć się o kilka kroków. `Ctrl+r` (redo) ponawia zmianę cofniętą przez `u`.

TYLKO ODCZYT

Powtórzenie ostatniej zmiany: .

```
.
```

Kropka powtarza ostatnią modyfikującą operację dokładnie w miejscu, w którym akurat stoi kursor. Przy powtarzalnej edycji w wielu miejscach pliku szybsze niż odtwarzanie całej sekwencji klawiszy za każdym razem.

🔒 TYLKO ODCZYT

Gramatyka: operator plus ruch

```
dw
d$
c$
dG
```

Operator sam w sobie nic nie robi, czeka na to, dokąd ma sięgnąć. `dw` znaczy usuń, przesuwając się jak `w`, czyli do początku następnego słowa. `d$` usuwa do końca linii, `c$` (change) usuwa do końca linii i od razu wchodzi w tryb wstawiania, `dG` usuwa od bieżącej linii do końca pliku.

Przykład:

```
3dw
```

Tabela operatorów i ruchów

Operator	Znaczenie	Przykładowy ruch	Wynik połączenia
`d`	usuń (delete)	`w`, `$`, `gg`, `G`	`dw` usuwa do końca słowa
`c`	zmień (change), usuwa i wchodzi w tryb wstawiania	`$`, `w`, `0`	`c$` zmienia resztę linii
`y`	kopiuj (yank)	`y`, `w`, `$`	`yy` kopiuje całą linię
`>`	zwiększ wcięcie	`>`, `G`	`>>` wcina bieżącą linię
`<`	zmniejsz wcięcie	`<`, `G`	`<<` cofa wcięcie bieżącej linii

Licznik przed całą komendą powtarza działanie: `3dw` usuwa trzy kolejne słowa, `2yy` kopiuje dwie linie.

Z życia wzięte

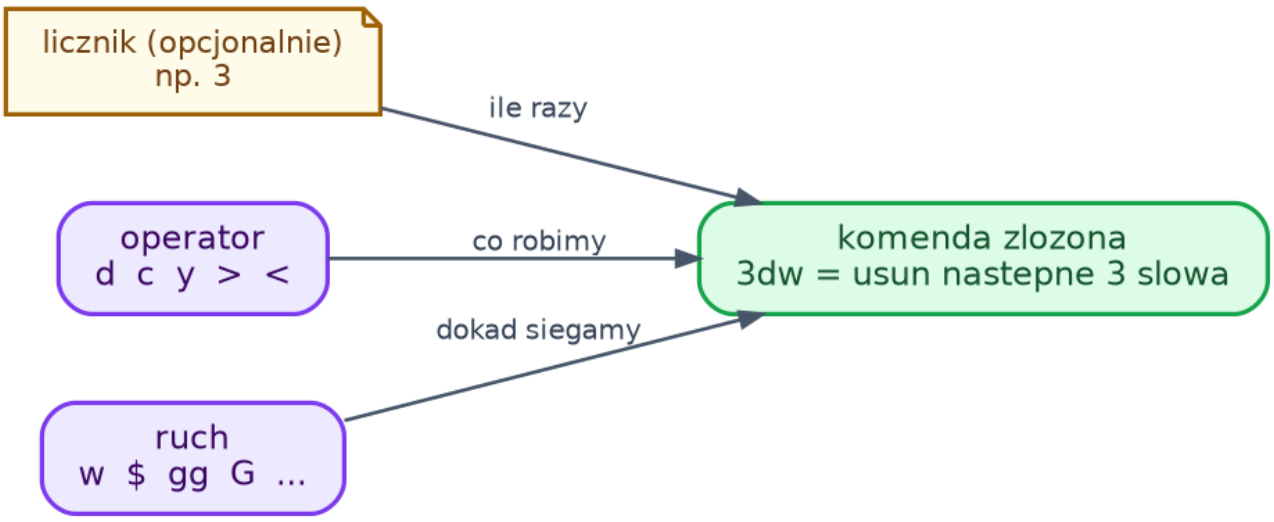
🔒 Z ŻYCIA WZIĘTE

> Ola prosi Tomka o usunięcie z pliku konfiguracyjnego pięciu kolejnych linii z przestarzałym ustawieniem. Tomek zamiast pięć razy naciskać `dd`, ustawia kursor na pierwszej z nich i wpisuje `5dd`, kasując wszystkie naraz jedną komendą.

Jak to działa

> **Pojęcie: bufor i rejestr.** Każda operacja kasowania (`dd`, `x`) i kopiowania (`yy`) odkłada usunięty albo skopiowany fragment do miejsca w pamięci edytora zwanego rejestrem. Domyślny, bezimienny rejestr działa jak schowek, z którego korzystają `p` i `P`. Kolejna operacja kasowania czy kopiowania nadpisuje ten sam rejestr: `dd` wykonane po `yy` bezpowrotnie zastępuje to, co skopiowaliśmy wcześniej. Do świadomego trzymania kilku fragmentów naraz służą rejestry nazwane literami, opisane na stronie Makra, rejestry, bloki i inne sztuczki.

Poniższy schemat pokazuje trzy elementy, z których składa się każda komenda edycji: opcjonalny licznik, operator i ruch.



Gramatyka Vima: licznik, operator i ruch łączą się w jedną komendę (kliknięcie otwiera pełny rozmiar)

Częste błędy

Objaw	Przyczyna	Co robimy
`dd` wykonane po `yy` wkleja nie to, co było skopiowane	oba operatory dzielą ten sam bezimienny rejestr, `dd` go nadpisało	do trzymania kilku fragmentów naraz używamy rejestrów nazwanych, patrz Makra, rejestry, bloki i inne sztuczki
Wpisanie `d` i nic się nie dzieje	`d` to operator, czeka na ruch, sam nic nie kasuje	dokańczamy komendę ruchem, na przykład `dw` albo `d$`, albo naciskamy `Esc`, żeby anulować
`.` powtarza coś innego, niż się spodziewaliśmy	`.` powtarza ostatnią modyfikującą operację, nie ostatnią komendę w ogóle, ruch bez zmiany tekstu się nie liczy	sprawdzamy, jaka była ostatnia rzeczywista zmiana tekstu przed użyciem `.`

Ćwiczenia

1. Chcemy usunąć cztery kolejne linie zaczynając od bieżącej, jedną komendą. Jak ją zapiszemy?

Odpowiedź

`4dd`. Licznik `4` przed `dd` powtarza kasowanie całej linii cztery razy, licząc od bieżącej w dół.

1. Na czym polega różnica między `dw` a `d$` wykonanymi w środku linii?

Odpowiedź

`dw` usuwa od kursora do początku następnego słowa, czyli fragment ograniczony ruchem `w`. `d$` usuwa od kursora do samego końca linii, niezależnie od tego, ile słów po drodze. Oba to ten sam operator `d`, różny tylko ruch, który do niego dołączymy.

Powiązane: Poruszanie się po pliku · Makra, rejestry, bloki i inne sztuczki · Szukanie i zamiana

Źródła: `man 1 vim` · `:help change.txt` w samym Vimie

Revision #30
Created 2026-09-27 11:19:46 UTC by Tomasz Konieczny
Updated 2026-10-26 14:07:24 UTC by Tomasz Konieczny