Makra, rejestry, bloki i inne sztuczki

Poziom: średni **Dotyczy: Vim 9.x** **Czas: ~25 min**

Powtarzalna edycja w wielu miejscach pliku nie musi oznaczać powtarzania tych samych klawiszy ręcznie. Vim nagrywa całe sekwencje jako makra, trzyma kilka fragmentów tekstu naraz w nazwanych rejestrach i edytuje prostokątne bloki kolumn jedną komendą.

Skróty użyte na tej stronie (1)

Skrót	Co znaczy
IP	Internet Protocol: adres komputera w sieci; w elektryce: stopień ochrony obudowy (Ingress Protection)

Fiszka

TYLKO ODCZYT

Nagrywanie makra: q

```
qa
...
q
```

`q` naciśnięte w trybie normalnym rozpoczyna nagrywanie makra, litera zaraz po nim (tu `a`) wskazuje rejestr, do którego trafi. Wszystko, co naciśniemy potem, jest zapamiętywane, aż do ponownego `q`, które kończy nagrywanie. To jedyne znaczenie litery `q` w Vimie, w odróżnieniu od `less` i `more`, gdzie ta sama litera kończy podgląd.

TYLKO ODCZYT

Odtworzenie makra: @a

```
@a
5@a
```

`@` plus litera rejestru odtwarza zapisaną w nim sekwencję klawiszy raz. Licznik przed komendą powtarza odtworzenie: `5@a` wykonuje makro pięć razy z rzędu.

□□ TYLKO ODCZYT

Powtórzenie ostatniego makra: @@

```
@@
```

Odtwarza ponownie makro, które było uruchomione jako ostatnie, bez podawania litery rejestru jeszcze raz.

□□ TYLKO ODCZYT

Rejestry nazwane: "a

```
"ayy
"ap
```

`"a` przed operacją kasowania albo kopiowania kieruje jej wynik do rejestru `a` zamiast do bezimiennego. `"ayy` kopiuje bieżącą linię do rejestru `a`, `"ap` wkleja z tego rejestru, niezależnie od tego, co w międzyczasie trafiło do rejestru bezimiennego przez zwykłe `dd` czy `yy`.

□□ TYLKO ODCZYT

Edycja blokowa: Ctrl+v i I

```
Ctrl+v
3j
I
#
Esc
```

`Ctrl+v` wchodzi w tryb wizualny blokowy, ruch (tu `3j`) zaznacza prostokątny blok kolumn w czterech kolejnych liniach, `I` (wielkie i) wstawia tekst na początku zaznaczenia we

wszystkich zaznaczonych liniach naraz, `#` to sam wpisywany znak, `Esc` zatwierdza operację na wszystkich liniach jednocześnie.

Przykład:

```
#linia1=a
#linia2=b
#linia3=c
#linia4=d
```

□□ TYLKO ODCZYT

Komenda globalna: :g

```
:g/wzorzec/d
```

`:g` (global) wykonuje wskazaną komendę na każdej linii pasującej do wzorca w całym pliku. `:g/wzorzec/d` kasuje od razu wszystkie linie zawierające dany tekst, bez ręcznego przechodzenia po nich.

□□ TYLKO ODCZYT

Sortowanie linii: :sort

```
:sort
:10,30sort
```

`:sort` sortuje alfabetycznie wszystkie linie pliku, `:10,30sort` tylko wskazany zakres. Przydatne przy porządkowaniu list nazw hostów czy adresów IP w pliku konfiguracyjnym.

□□ **IP** = Internet Protocol: adres komputera w sieci; w elektryce: stopień ochrony obudowy (Ingress Protection)

□□ TYLKO ODCZYT

Uruchomienie komendy powłoki: :!komenda

```
:!ls -la
```

Uruchamia wskazaną komendę powłoki, pokazuje jej wynik i wraca do edytora bez zmiany bufora. Różni się tym od `:r !komenda`, które wynik wstawia do tekstu w miejscu kursora.

◫◫ ZMIENIA SYSTEM

Wcięcia: >> i <<

```
>>
<<
3>>
```

`>>` zwiększa wcięcie bieżącej linii o jeden poziom, `<<` zmniejsza. Licznik przed komendą obejmuje kolejne linie: `3>>` wcina trzy linie licząc od bieżącej.

Z życia wzięte

◫◫ Z ŻYCIA WZIĘTE

Ola musi zakomentować dwadzieścia linii starych ustawień w pliku `nginx.conf` przed testem nowej konfiguracji. Zamiast wchodzić w tryb wstawiania dwadzieścia razy, nagrywa makro: `qa`, `I`, `#`, `Esc`, `j`, `q`, a potem odtwarza je dziewiętnaście razy komendą `19@a`, kończąc całą operację w kilka sekund.

Jak to działa

Pojęcie: makro. Nagrana sekwencja klawiszy przypisana do jednego rejestru, którą Vim odtwarza dokładnie tak, jakbyśmy sami naciskali te same klawisze jeszcze raz, tylko szybciej i bez pomyłek. Makro nie rozumie treści pliku, powtarza ruchy ślepo, więc działa najlepiej tam, gdzie kolejne linie mają tę samą strukturę.

Ta sama komenda komentowania wielu linii, którą stara strona autora uczyła przez zakres i wyrażenie regularne, działa też przez tryb blokowy:

Sposób	Komenda	Kiedy wygodniejszy
Zakres i podstawienie	`:10,20s/^/#/`	znamy dokładne numery linii, nie chcemy zaznaczać ręcznie

Sposób	Komenda	Kiedy wygodniejszy
Tryb blokowy	`Ctrl+v`, ruch w dół, `I`, `#`, `Esc`	widzimy zakres na ekranie, wygodniej zaznaczyć niż liczyć linie

Odkomentowanie działa lustrzanie: zakresem przez `:10,20s/^#//`, albo blokowo przez zaznaczenie kolumny ze znakami `#` w trybie `Ctrl+v` i skasowanie jej przez `x`.

Częste błędy

Objaw	Przyczyna	Co robimy
Naciśnięcie `q` w Vimie licząc, że zamknie plik jak w `less`	`q` w Vimie zaczyna nagrywanie makra, nie jest komendą wyjścia	<kbd>Esc</kbd>, potem `:q`, `:wq` albo `:q!`, patrz Pierwsze kroki: otwarcie, zapis, wyjście
Makro zaczyna nagrywać, mimo że nie chcieliśmy	przypadkowe naciśnięcie `q` i litery rejestru w trybie normalnym	`q` jeszcze raz kończy nagrywanie, potem `u` cofa niechciane zmiany wprowadzone w trakcie
`I` w trybie blokowym wstawia tekst tylko w pierwszej linii zaznaczenia	zaznaczenie zatwierdzono przed wpisaniem znaku, na przykład przez Enter zamiast bezpośredniego wpisania litery	powtarzamy sekwencję, kończąc ją od razu klawiszem <kbd>Esc</kbd> po wpisanym znaku

□□ Ćwiczenia

1. Mamy dwadzieścia linii do odkomentowania, wszystkie zaczynają się znakiem `#`. Jak zrobić to jedną komendą zakresową?

Odpowiedź

`:1,20s/^#//` (albo dopasowany zakres do faktycznych numerów linii). Wyrażenie `^#` dopasowuje znak `#` na początku linii, puste zastąpienie go usuwa.

1. Nagraliśmy makro do rejestru `b` i chcemy odtworzyć je dziesięć razy z rzędu. Jaka jest najkrótsza komenda?

Odpowiedź

`10@b`. Licznik przed `@` i literą rejestru powtarza odtworzenie makra tyle razy, ile wskazuje liczba, bez wpisywania `@b` po kolei dziesięć razy.

Powiązane: Edycja: kasowanie, kopiowanie, wklejanie, cofanie · Szukanie i zamiana · Tryby: dlaczego klawisze robią różne rzeczy

Źródła: `man 1 vim` · `:help repeat.txt`, `:help usr_10.txt`, `:help change.txt` w samym Vimie

Revision #31
Created 2025-03-07 13:21:42 UTC by Tomasz Konieczny
Updated 2026-10-26 14:07:28 UTC by Tomasz Konieczny