

1. Czym jest VIM ?
2. Czym jest VI ?
3. Jakie występują tryby pracy programu i jak się je uruchamia?
4. Jakie są sposoby nawigacji po między znakami, wyrazami, wierszami?
5. Jak kopiować fragment tekstu i wklejać go dowolnie ?
6. Znajdź kilka stron poświęconych tematyce VIMa oraz obsługi programu (może być gra)
7. Jak uruchomić podręcznik w programie VIM
 - a. Uruchomić Tutorial i wypełnić go w całości
8. Jakie są inne edytory tekstu w Linuxie
 - a. Wbudowane
 - b. Zainstalować min 5 dodatkowych (pokaż ich działanie)
9. Utworzyć w programie VIM dokument tekstowy z tekstem piosenki (ona tu jest i tańczy dla mnie...)
10. Usunąć 4 wiersz następnie jeszcze 7 wyraz
11. Zmienić wielkość liter (z dużych na małe i odwrotnie) 5 szt
12. Skopiować tekst „Ja uwielbiam ją” za pomocą polecenia a następnie wklejenie go 10 razy
13. Słowo „ją” zamienić na „placki”
14. Oznaczyć słowo placki w dokumencie (wyświetl efekt i polecenie)
15. Posortować cały tekst
16. Posortować tekst odwrotnie
17. Zapisać cały dokument pod swoim nazwiskiem na pulpicie (pokaż działanie)
18. Wczytać tekst za pomocą terminalu i wyświetlenie zawartości

<http://leniwy.eu/news,5,Podstawy-edytora-Vim-skroty-klawiszowe.html>

http://computersun.pl/linux/podstawy/vim-podstawowy-edytor-tekstu-s_96.html

<http://vim.cba.pl/>

<http://www.openvim.com/tutorial.html>

<http://www.arturpyszczuk.pl/vimtutorial.html>

[Obsługa edytora tekstu VIM \[KoNaR\] - YouTube](#)

[Alfabet VIMa - czyli jak zrozumieć ten edytor? - YouTube](#)

[Vim - Zaawansowany edytor tekstu | chyla.org](#)

[Interactive Vim tutorial \(openvim.com\)](#)

dla tych którym nie działa z jakichś powodów VIM podaje niżej treść Vimtutora

+++++

=====
=====

= W i t a j w t u t o r i a l u V I
M - a - Wersja 1.7. =

=====
=====

Vim to potężny edytor, który posiada wiele poleceń, zbyt dużo, by

wyjaśnić je wszystkie w tym tutorialu.

Ten przewodnik ma nauczyć

Cię **pos**ługiwać się wystarczająco

wieloma komendami, byś **mó**gł łatwo

używać Vima jako edytora **ogól**nego

przeznaczenia.

Czas potrzebny na ukończenie tutoriala to 25 **do** 30 minut i zależy

od tego jak wiele czasu spędzisz na eksperymentowaniu.

UWAGA:

Polecenia wykonywane w czasie lekcji zmodyfikują tekst. Zrób

wcześniej kopię tego pliku **do**

ćwiczeń (jeśli zacząłeś komendą

"vimtutor", to już pracujesz na

kopii).

Pamiętaj, że przewodnik ten został zaprojektowany **do** nauki poprzez

ćwiczenia. Oznacza to, że musisz

wykonywać polecenia, by nauczyć się ich

prawidłowo. Jeśli będziesz jedynie

czytał tekst, szybko zapomnisz wiele

poleceń!

Teraz upewnij się, że nie masz wciśniętego CapsLocka i wciskaj **j**

tak długo dopóki Lekcja 1.1. nie

wypełni całkowicie ekranu.

~~~~~  
~~~~~

Lekcja 1.1.: PORUSZANIE SIĘ

KURSOREM

** By wykonać ruch kursorem, wciśnij h, j, k, l jak pokazano. **

^

k

Wskazówka: h jest po lewej

< h l >

l jest po prawej

j

j

wygląda jak strzałka w dół

v

1. Poruszaj kursorem dopóki nie będziesz pewien, że pamiętasz polecenia.

2. Trzymaj j tak długo aż będzie się powtarzał.

Teraz wiesz jak dojść do następnej lekcji.

3. Używając strzałki w dół przejdź do następnej lekcji.

Uwaga: Jeśli nie jesteś pewien czegoś co wpisałeś, wciśnij <ESC>, by wrócić do trybu Normal. Wtedy powtórz polecenie.

Uwaga: Klawisze kursora także powinny działać, ale używając hjkl będziesz w stanie poruszać się o wiele szybciej, jak się tylko przyzwyczaisz. Naprawdę!

~~~~~  
~~~~~

Lekcja 1.2.: WYCHODZENIE Z VIM-a

!! UWAGA: Przed wykonaniem jakiegokolwiek polecenia przeczytaj całą lekcję !!

1. Wciśnij <ESC> (aby upewnić się, że jesteś w trybie Normal).

2. Wpisz: :q!<ENTER>.

To spowoduje wyjście z edytora

PORZUCAJĄC wszelkie zmiany, jakie
zdażyłeś zrobić. Jeśli chcesz
zapamiętać zmiany i wyjść,
wpisz: :wq<ENTER>

3. Kiedy widzisz znak zachęty powłoki
wpisz komendę, żeby wrócić
do tutoriala. Czyli: vimtutor<ENTER>

4. Jeśli chcesz zapamiętać polecenia,
wykonaj kroki 1. do 3., aby
wyjść i wrócić do edytora.

UWAGA: :q!<ENTER> porzuca wszelkie zmiany
jakie zrobiłeś. W następnych
lekcjach dowiesz się jak je
zapamiętywać.

5. Przenieś kursor do lekcji 1.3.

~~~~~  
~~~~~

Lekcja 1.3.: EDYCJA TEKSTU -
KASOWANIE

** Wciśnij x aby usunąć znak
pod kursorem. **

1. Przenieś kursor do linii poniżej
oznaczonej --->.

2. By poprawić błędy, naprowadź kursor na
znak do usunięcia.

3. Wciśnij x aby usunąć niechciany znak.

4. Powtarzaj kroki 2. do 4. dopóki zdanie
nie jest poprawne.

--->Kkrowaprrzeskoczyłaprrzezksiiężycc.

5. Teraz, kiedy zdanie jest poprawione,
przejdź do Lekcji 1.4.

UWAGA: Ucz się przez ćwiczenie, nie

wkuwanie.

~~~~~  
~~~~~  
Lekcja 1.4.: EDYCJA TEKSTU -
INSERT (wprowadzanie)

** Wciśnij i aby wstawić
tekst. **

1. Przenieś kursor do pierwszej linii
poniżej oznaczonej --->.

2. Aby poprawić pierwszy wiersz, ustaw
kursor na pierwszym znaku PO tym,
gdzie tekst ma być wstawiony.

3. Wciśnij i a następnie wpisz konieczne
poprawki.

4. Po poprawieniu błędu wciśnij <ESC>, by
wrócić do trybu Normal.

Powtarzaj kroki 2. do 4., aby poprawić
całe zdanie.

---> W tej brkje trochę .

---> W tej linii brakuje trochę tekstu.

5. Kiedy czujesz się swobodnie wstawiając
tekst, przejdź do
podsumowania poniżej.

~~~~~  
~~~~~  
Lekcja 1.5.: EDYCJA TEKSTU -
APPENDING (dodawanie)

** Wciśnij A by dodać
tekst. **

1. Przenieś kursor **do** pierwszej linii poniżej oznaczonej --->.

Nie ma znaczenia, który to będzie znak.

2. Wciśnij **A** i wpisz odpowiednie dodatki.

3. Kiedy tekst został dodany, wciśnij <ESC> i wróć **do** trybu Normalnego.

4. Przenieś kursor **do** drugiej linii oznaczonej ---> i powtórz kroki 2. i 3., aby poprawić zdanie.

---> Brakuje tu tro
Brakuje tu trochę tekstu.

---> Tu też trochę bra
Tu też trochę brakuje.

5. Kiedy już utrwaliłeś ćwiczenie, przejdź **do** lekcji 1.6.

~~~~~  
~~~~~

Lekcja 1.6.: EDYCJA

PLIKU

**** Użyj :wq aby zapisać plik i wyjść. ****

!! UWAGA: zanim wykonasz jakiegokolwiek polecenia przeczytaj całą lekcję !!

1. Zakończ tutorial tak jak w lekcji 1.2.:
:q!

lub, jeśli masz dostęp do innego terminala, wykonaj kolejne kroki tam.

2. W powłoce wydaj polecenie: vim
tutor<ENTER>

"vim" jest poleceniem uruchamiającym edytor Vim. 'tutor' to nazwa pliku, jaki chcesz edytować. Użyj pliku, który może zostać zmieniony.

3. Dodaj i usuń tekst tak, jak się nauczyłeś w poprzednich lekcjach.

4. Zapisz plik ze zmianami i opuść
Vima: :wq<ENTER>

5. Jeśli zakończyłeś vimtutor w kroku 1., uruchom go ponownie i przejdź do podsumowania poniżej.

6. Po przeczytaniu wszystkich kroków i ich zrozumieniu: wykonaj je.

~~~~~  
~~~~~

LEKCJA 1.

PODSUMOWANIE

1. Poruszasz kursorem używając "strzałek" i klawiszy hjkl .
h (w lewo) j (w dół) k (do góry)
l (w prawo)

2. By wejść do Vima, (z powłoki) wpisz:
vim NAZWA_PLIKU<ENTER>

3. By wyjść z Vima, wpisz:
<ESC> :q!<ENTER> by usunąć wszystkie zmiany.
LUB: <ESC> :wq<ENTER> by zmiany zachować.

4. By usunąć znak pod kursorem, wciśnij:
x

5. By wstawić tekst przed kursorem lub dodać:
i wpisz tekst <ESC>
wstawi przed kursorem
A wpisz tekst <ESC> doda
na końcu linii

UWAGA: Wciśnięcie <ESC> przeniesie Cię z powrotem do trybu Normal lub odwoła niechciane lub częściowo wprowadzone polecenia.

Teraz możemy kontynuować i przejść do Lekcji 2.

~~~~~  
~~~~~

Lekcja 2.1.: POLECENIE
DELETE (usuwanie)

** Wpisz dw by usunąć
wyraz. **

1. Wciśnij <ESC>, by upewnić się, że
jesteś w trybie Normal.

2. Przenieś kursor do linii poniżej
oznaczonej --->.

3. Przesuń kursor na początek wyrazu,
który chcesz usunąć.

4. Wpisz dw by usunąć wyraz.

UWAGA: Litera d pojawi się na dole
ekranu. Vim czeka na wpisanie w .

Jeśli zobaczysz inny znak, oznacza
to, że wpisałeś coś źle; wciśnij
<ESC> i zacznij od początku.

---> Jest tu parę papier wyrazów, które
kamień nie należą donożyce tego zdania.

5. Powtarzaj kroki 3. i 4. dopóki zdanie
nie będzie poprawne, potem
przejdź do Lekcji 2.2.

~~~~~  
~~~~~

Lekcja 2.2.: WIĘCEJ
POLECEŃ USUWAJĄCYCH

** Wpisz d\$ aby usunąć tekst
do końca linii. **

1. Wciśnij <ESC> aby się upewnić, że

jesteś w trybie Normal.

2. Przenieś kursor **do** linii poniżej
oznaczonej --->.

3. Przenieś kursor **do** końca poprawnego
zdania (PO pierwszej .).

4. Wpisz **d\$** aby usunąć resztę linii.

---> Ktoś wpisał koniec tego zdania
dwukrotnie. zdania dwukrotnie.

5. Przejdź **do** Lekcji 2.3., by zrozumieć co
się stało.

~~~~~  
~~~~~

Lekcja 2.3.: 0

OPERATORACH I RUCHACH

Wiele poleceń zmieniających tekst jest
złożonych z operatora i ruchu.

Format dla polecenia usuwającego z
operatorem **d** jest następujący:

d **ruch**

gdzie:

d - operator usuwania.

ruch - na czym polecenie będzie
wykonywane (lista poniżej).

Krótką listą ruchów:

w - **do** początku następnego wyrazu
WYŁĄCZAJĄC pierwszy znak.

e - **do** końca bieżącego wyrazu, WŁĄCZAJĄC
ostatni znak.

\$ - **do** końca linii, WŁĄCZAJĄC ostatni
znak.

W ten sposób wpisanie `de` usunie znaki od kursora `do` końca wyrazu.

UWAGA: Wpisanie tylko ruchu w trybie Normal bez operatora przeniesie kursor tak, jak to określono.

~~~~~  
~~~~~

Lekcja 2.4.: UŻYCIE MNOŻNIKA DLA RUCHU

** Wpisanie liczby przed ruchem powtarza ruch odpowiednią ilość razy. **

1. Przenieś kursor na początek linii poniżej zaznaczonej `-->`.

2. Wpisz `2w` aby przenieść kursor o dwa wyrazy `do` przodu.

3. Wpisz `3e` aby przenieść kursor `dokońca` trzeciego wyrazu w przód.

4. Wpisz `0` (zero), aby przenieść kursor na początek linii.

5. Powtórz kroki 2. i 3. z innymi liczbami.

`-->` To jest zwykły wiersz z wyrazami, po których możesz się poruszać.

6. Przejdź `do` lekcji 2.5.

~~~~~  
~~~~~

Lekcja 2.5.: UŻYCIE MNOŻNIKA, BY WIĘCEJ USUNĄĆ

** Wpisanie liczby z operatorem powtarza go odpowiednią ilość razy. **

W wyżej wspomnianej kombinacji operatora usuwania i ruchu podaj mnożnik przed ruchem, by więcej usunąć:
d liczba ruch

1. Przenieś kursor **do** pierwszego wyrazu KAPITALIKAMI w linii zaznaczonej --->.

2. Wpisz 2dw aby usunąć dwa wyrazy KAPITALIKAMI.

3. Powtarzaj kroki 1. i 2. z innymi mnożnikami, aby usunąć kolejne wyrazy KAPITALIKAMI jednym poleceniem

---> ta ASD WE linia QWE ASDF ZXCVC FG wyrazów została ERT FGH CF oczyszczona.

UWAGA: Mnożnik pomiędzy operatorem d i ruchem działa podobnie **do** ruchu bez operatora.

~~~~~  
~~~~~

Lekcja 2.6.: OPEROWANIE NA LINIACH

** Wpisz dd aby usunąć całą linię. **

Z powodu częstości usuwania całych linii, projektanci Vi zdecydowali, że będzie łatwiej wpisać dwa razy d aby usunąć linię.

1. Przenieś kursor **do** drugiego zdania z wierszyka poniżej.

2. Wpisz dd aby usunąć wiersz.

3. Teraz przenieś się **do** czwartego wiersza.

4. Wpisz 2dd aby usunąć dwa wiersze.

- > 1) Róże są czerwone,
- > 2) Błoto jest fajne,
- > 3) Fiołki są niebieskie,
- > 4) Mam samochód,
- > 5) Zegar podaje czas,
- > 6) Cukier jest słodki,
- > 7) I ty też.

~~~~~  
~~~~~

Lekcja 2.7.: POLECENIE

UNDO (cofnij)

** Wciśnij u aby cofnąć skutki ostatniego polecenia.

U zaś, by cofnąć skutki dla całej linii. **

1. Przenieś kursor do zdania poniżej oznaczonego ---> i umieść go na pierwszym błędzie.
2. Wpisz x aby usunąć pierwszy niechciany znak.
3. Teraz wciśnij u aby cofnąć skutki ostatniego polecenia.
4. Tym razem popraw wszystkie błędy w linii używając polecenia x .
5. Teraz wciśnij wielkie U aby przywrócić linię do oryginalnego stanu.
6. Teraz wciśnij u kilka razy, by cofnąć U i poprzednie polecenia.
7. Teraz wpisz CTRL-R (trzymaj równocześnie wciśnięte klawisze CTRL i R) kilka razy, by cofnąć cofnięcia.

--->Popraw błędy w tej linii i zamień je przezcofnij.

8. To są bardzo przyteczne polecenia.

Przejdź teraz do podsumowania Lekcji 2.

~~~~~

~~~~~  
LEKCJA 2.

PODSUMOWANIE

1. By usunąć znaki od kursora do następnego wyrazu, wpisz: `dw`
2. By usunąć znaki od kursora do końca linii, wpisz: `d$`
3. By usunąć całą linię: `dd`
4. By powtórzyć ruch, poprzedź go liczbą:
`2w`
5. Format polecenia zmiany to:
`operator [liczba] ruch`
gdzie:
`operator` - to, co trzeba zrobić (np. `d` dla usuwania)
`[liczba]` - opcjonalne, ile razy powtórzyć ruch
`ruch` - przenosi nad tekstem do operowania, takim jak `w` (wyraz),
`$` (do końca linii) etc.
6. By przejść do początku linii, użyj zera: `0`
7. By cofnąć poprzednie polecenie, wpisz:
`u` (małe u)
By cofnąć wszystkie zmiany w linii, wpisz: `U` (wielkie U)
By cofnąć cofnięcie, wpisz:
`CTRL-R`

~~~~~  
~~~~~  
Lekcja 3.1.:

POLECENIE PUT (wstaw)

** Wpisz `p` by wstawić ostatnie usunięcia za kursorem. **

1. Przenieś kursor do pierwszej linii ---> poniżej.

2. Wpisz dd aby usunąć linię i przechować ją w rejestrze Vima.

3. Przenieś kursor do linii c), POWYŻEJ tej, gdzie usunięta linia powinna się znajdować.

4. Wciśnij p by wstawić linię poniżej kursora.

5. Powtarzaj kroki 2. do 4. aż znajdą się w odpowiednim porządku.

---> d) Jak dwa aniołki.

---> b) Na dole fiołki,

---> c) A my się kochamy,

---> a) Na górze róże,

~~~~~  
~~~~~

Lekcja 3.2.: POLECENIE

REPLACE (zastąp)

** Wpisz rx aby zastąpić znak pod kursorem na x . **

1. Przenieś kursor do pierwszej linii poniżej oznaczonej --->

2. Ustaw kursor na pierwszym błędzie.

3. Wpisz r a potem znak jaki powinien go zastąpić.

4. Powtarzaj kroki 2. i 3. dopóki pierwsza linia nie będzie taka, jak druga.

--->Kiedy ten wiersz był wstukiwany, ktoś wcisnął parę złych klawirzy!

---> Kiedy ten wiersz był wstukiwany, ktoś wcisnął parę złych klawiszy!

5. Teraz czas na Lekcję 3.3.

UWAGA: Pamiętaj, by uczyć się ćwicząc, a nie pamięciowo.

~~~~~  
~~~~~

Lekcja 3.3.: OPERATOR

CHANGE (zmień)

** By zmienić do końca
wyrazu, wpisz ce . **

1. Przenieś kursor do pierwszej linii
poniżej oznaczonej --->.

2. Umieść kursor na u w lunos.

3. Wpisz ce i popraw wyraz (w tym
wypadku wstaw inia).

4. Wciśnij <ESC> i przejdź do następnej
planowanej zmiany.

5. Powtarzaj kroki 3. i 4. dopóki pierwsze
zdanie nie będzie takie same,
jak drugie.

---> Ta lunos ma pirc słów, które
tżinazbnicużifajoncpcmazu zmien.

---> Ta linia ma parę słów, które trzeba
zmienić używając polecenia zmien.

Zauważ, że ce nie tylko zamienia wyraz,
ale także zmienia tryb na
Insert (wprowadzanie).

~~~~~  
~~~~~

Lekcja 3.4.: WIĘCEJ

ZMIAN UŻYWAJĄC c

** Polecenie change używa takich
samych ruchów, jak delete. **

1. Operator change działa tak samo, jak delete. Format wygląda tak:

c [liczba] ruch

2. Ruchy są także takie same, np.: w (wyraz), \$ (koniec linii) etc.

3. Przenieś się do pierwszej linii poniżej oznaczonej --->

4. Ustaw kursor na pierwszym błędzie.

5. Wpisz c\$, popraw koniec wiersza i wciśnij <ESC>.

---> Koniec tego wiersza musi być poprawiony, aby wyglądał tak, jak drugi.
---> Koniec tego wiersza musi być poprawiony używając polecenia c\$.

UWAGA: Możesz używać <BS> aby poprawiać błędy w czasie pisania.

~~~~~  
~~~~~

LEKCJA 3.

PODSUMOWANIE

1. Aby wstawić tekst, który został wcześniej usunięty wciśnij p . To polecenie wstawia skasowany tekst PO kursorze (jeśli cała linia została usunięta, zostanie ona umieszczona w linii poniżej kursora).

2. By zamienić znak pod kursorem, wciśnij r a potem znak, który ma zastąpić oryginalny.

3. Operator change pozwala Ci na zastąpienie od kursora do miejsca, gdzie zabrałby Cię ruch. Np. wpisz ce aby zamienić tekst od kursora do końca

wyrazu, c\$ aby zmienić tekst do końca linii.

4. Format do polecenia change (zmień):

c [liczba] obiekt

Teraz przejdź do następnej lekcji.

~~~~~  
~~~~~

Lekcja 4.1.: POŁOŻENIE KURSORA ORAZ STATUS PLIKU

** Naciśnij CTRL-G aby zobaczyć swoje położenie w pliku i status pliku. Naciśnij G aby przejść do linii w pliku. **

UWAGA: Przeczytaj całą lekcję zanim wykonasz jakieś polecenia!!!

1. Przytrzymaj klawisz CTRL i wciśnij g . Używamy notacji CTRL-G.

Na dole strony pojawi się pasek statusu z nazwą pliku i pozycją w pliku.

Zapamiętaj numer linii dla potrzeb kroku 3.

UWAGA: Możesz też zobaczyć pozycję kursora w prawym, dolnym rogu ekranu.

Dzieje się tak kiedy ustawiona jest opcja 'ruler' (więcej w lekcji 6.).

2. Wciśnij G aby przejść na koniec pliku.

Wciśnij gg aby przejść do początku pliku.

3. Wpisz numer linii, w której byłeś a potem G . To przeniesie Cię

z powrotem do linii, w której byłeś kiedy wcisnąłeś CTRL-G.

4. Jeśli czujesz się wystarczająco pewnie,

wykonaj kroki 1-3.

~~~~~  
~~~~~

Lekcja 4.2.:

POLECENIE SZUKAJ

** Wpisz / a następnie
wyrażenie, aby je znaleźć. **

1. W trybie Normal wpisz / . Zauważ, że
znak ten oraz kursor pojawia
się na dole ekranu tak samo, jak
polecenie : .

2. Teraz wpisz blond<ENTER> . To jest
słowo, którego chcesz szukać.

3. By szukać tej samej frazy ponownie, po
prostu wciśnij n .

Aby szukać tej frazy w przeciwnym,
kierunku wciśnij N .

4. Jeśli chcesz szukać frazy do tyłu, użyj
polecenia ? zamiast / .

5. Aby wrócić gdzie byłeś, wciśnij CTRL-
O. Powtarzaj, by wrócić dalej. CTRL-I
idzie do przodu.

Uwaga: 'blond' to nie jest metoda, by
przeliterować błąd; 'blond' to błąd.

Uwaga: Kiedy szukanie osiągnie koniec
pliku, będzie kontynuowane od początku

o ile opcja 'wrapscan' nie została
przestawiona.

~~~~~  
~~~~~

Lekcja 4.3.: W POSZUKIWANIU

PARUJĄCYCH NAWIASÓW

** Wpisz % by znaleźć
parujący),], lub } . **

1. Umieść kursor na którymś z (, [, lub { w linii poniżej oznaczonej --->.
2. Teraz wpisz znak % .
3. Kursor powinien się znaleźć na parującym nawiasie.
4. Wciśnij % aby przenieść kursor z powrotem do parującego nawiasu.
5. Przenieś kursor do innego (,), [,], { lub } i zobacz co robi % .

---> To (jest linia testowa z (, [,] i {, } .))

Uwaga: Ta funkcja jest bardzo użyteczna w debuggowaniu programu z niesparowanymi nawiasami!

~~~~~  
~~~~~

Lekcja 4.4.: POLECENIE SUBSTITUTE (zamiana)

** Wpisz :s/stary/nowy/g aby zamienić 'stary' na 'nowy'. **

1. Przenieś kursor do linii poniżej oznaczonej --->.
2. Wpisz :s/czaas/czas<ENTER> . Zauważ, że to polecenie zmienia tylko pierwsze wystąpienie 'czaas' w linii.
3. Teraz wpisz :s/czaas/czas/g . Dodane g oznacza zamianę (substytucję) globalnie w całej linii. Zmienia wszystkie wystąpienia 'czaas' w linii.

---> Najlepszy czas na zobaczenie najładniejszych kwiatów to czas wiosny.

4. Aby zmienić wszystkie wystąpienia łańcucha znaków pomiędzy dwoma liniami, wpisz: `:#,#s/stare/nowe/g` gdzie #,# są numerami linii ograniczających region, gdzie ma nastąpić zamiana.

wpisz `:%s/stare/nowe/g` by zmienić wszystkie wystąpienia w całym pliku.

wpisz `:%s/stare/nowe/gc` by zmienić wszystkie wystąpienia w całym pliku, prosząc o potwierdzenie za każdym razem.

~~~~~  
~~~~~

LEKCJA 4.

PODSUMOWANIE

1. CTRL-G pokaże Twoją pozycję w pliku i status pliku. SHIFT-G przenosi Cię do końca pliku.
G przenosi do końca pliku.
liczba G przenosi do linii [liczba].
gg przenosi do pierwszej linii.

2. Wpisanie `/` a następnie łańcucha znaków szuka łańcucha DO PRZODU.

Wpisanie `?` a następnie łańcucha znaków szuka łańcucha DO TYŁU.

Po wyszukiwaniu wciśnij `n` by znaleźć następne wystąpienie szukanej

frazy w tym samym kierunku lub `N` by szukać w kierunku przeciwnym.

CTRL-O przenosi do starszych pozycji, CTRL-I do nowszych.

3. Wpisanie `%` gdy kursor znajduje się na `(,), [,], {, }` lub `}` lokalizuje parujący znak.

4. By zamienić pierwszy stary na nowy w linii, wpisz `:s/stary/nowy`

By zamienić wszystkie stary na nowy w

linii, wpisz :s/stary/nowy/g
By zamienić frazy pomiędzy dwoma
liniami # wpisz :#,#s/stary/nowy/g
By zamienić wszystkie wystąpienia w
pliku, wpisz :%s/stary/nowy/g
By Vim prosił Cię o potwierdzenie,
dodaj 'c' :%s/stary/nowy/gc

~~~~~  
~~~~~

Lekcja 5.1.: JAK WYKONAĆ POLECENIA ZEWNĘTRZNE?

** Wpisz :! a następnie zewnętrzne
polecenie, by je wykonać. **

1. Wpisz znajome polecenie : by ustawić
kursor na dole ekranu. To pozwala
na wprowadzenie komendy linii poleceń.

2. Teraz wstaw ! (wykrzyknik). To
umożliwi Ci wykonanie dowolnego
zewnętrznego polecenia powłoki.

3. Jako przykład wpisz ls za ! a
następnie wciśnij <ENTER>. To polecenie
pokaże spis plików w Twoim katalogu,
tak jakbyś był przy znaku zachęty
powłoki. Możesz też użyć :!dir jeśli
ls nie działa.

Uwaga: W ten sposób można wykonać wszystkie
polecenia powłoki.

Uwaga: Wszystkie polecenia : muszą być
zakończone <ENTER>.

Od tego momentu nie zawsze będziemy
o tym wspominać.

~~~~~  
~~~~~

Lekcja 5.2.: WIĘCEJ O ZAPISYWANIU PLIKÓW

**** By zachować zmiany w tekście,
wpisz :w NAZWA_PLIKU . ****

1. Wpisz `!dir` lub `!ls` by zobaczyć spis plików w katalogu.

Już wiesz, że musisz po tym wcisnąć `<ENTER>`.

2. Wybierz nazwę pliku, jaka jeszcze nie istnieje, np. TEST.

3. Teraz wpisz: `:w TEST` (gdzie TEST jest nazwą pliku jaką wybrałeś.)

4. To polecenie zapamięta cały plik (Vim Tutor) pod nazwą TEST.

By to sprawdzić, wpisz `!dir` lub `!ls` żeby znowu zobaczyć listę plików.

Uwaga: Zauważ, że gdybyś teraz wyszedł z Vima, a następnie wszedł ponownie poleceniem `vim TEST`, plik byłby dokładną kopią tutoriala, kiedy go zapisywałeś.

5. Teraz usuń plik wpisując (MS-DOS):

`!del TEST`

lub (Unix):

`!rm TEST`

~~~~~  
~~~~~

Lekcja 5.3.: WYBRANIE
TEKSTU DO ZAPISU

**** By zachować część pliku, wpisz
v ruch :w NAZWA_PLIKU ****

1. Przenieś kursor do tego wiersza.

2. Wciśnij `v` i przenieś kursor do punktu

5. Zauważ, że tekst został podświetlony.

3. Wciśnij znak `:`. Na dole ekranu pojawi się `:'<,'>`.

4. Wpisz `w` w `TEST`, gdzie `TEST` to nazwa pliku, który jeszcze nie istnieje.

Upewnij się, że widzisz `:'<,'>` w `TEST` zanim wciśniesz `Enter`.

5. Vim zapisze wybrane linie do pliku `TEST`. Użyj `!dir` lub `!ls`, żeby to zobaczyć. Jeszcze go nie usuwaj! Użyjemy go w następnej lekcji.

UWAGA: Wciśnięcie `v` zaczyna tryb Wizualny. Możesz poruszać kursorem, by zmienić rozmiary zaznaczenia. Możesz też użyć operatora, by zrobić coś z tekstem. Na przykład `d` usuwa tekst.

~~~~~  
~~~~~

Lekcja 5.4.: WSTAWIANIE I ŁĄCZENIE PLIKÓW

**** By wstawić zawartość pliku, wpisz `:r NAZWA_PLIKU` ****

1. Umieść kursor tuż powyżej tej linii.

UWAGA: Po wykonaniu kroku 2. zobaczysz tekst z Lekcji 5.3. Potem przejdź do `DOŁU`, by zobaczyć ponownie tę lekcję.

2. Teraz wczytaj plik `TEST` używając polecenia `:r TEST`, gdzie `TEST` jest nazwą pliku. Wczytany plik jest umieszczony poniżej linii z kursorem.

3. By sprawdzić czy plik został wczytany, cofnij kursor i zobacz, że teraz są dwie kopie Lekcji 5.3., oryginał i kopia z pliku.

UWAGA: Możesz też wczytać wyjście
zewnętrznego polecenia. Na przykład
:r !ls wczytuje wyjście polecenia ls
i umieszcza je pod poniżej
kursora.

~~~~~  
~~~~~

LEKCJA 5.

PODSUMOWANIE

1. :!polecenie wykonuje polecenie
zewnętrzne.

Użytecznymi przykładami są:

:!dir - pokazuje spis plików w
katalogu.

:!rm NAZWA_PLIKU - usuwa plik
NAZWA_PLIKU.

2. :w NAZWA_PLIKU zapisuje obecny plik
Vima na dysk z nazwą NAZWA_PLIKU.

3. v ruch :w NAZWA_PLIKU zapisuje
Wizualnie wybrane linie do NAZWA_PLIKU.

4. :r NAZWA_PLIKU wczytuje z dysku plik
NAZWA_PLIKU i wstawia go do
bieżącego pliku poniżej kursora.

5. :r !dir wczytuje wyjście polecenia
dir i umieszcza je poniżej kursora.

~~~~~  
~~~~~

Lekcja 6.1.: POLECENIE

OPEN (otwórz)

** Wpisz o by otworzyć linię poniżej kursora i przenieść się do trybu Insert (wprowadzanie). **

1. Przenieś kursor do linii poniżej oznaczonej --->.

2. Wpisz o (małe), by otworzyć linię PONIŻEJ kursora i przenieść się do trybu Insert (wprowadzanie).

3. Wpisz trochę tekstu i wciśnij <ESC> by wyjść z trybu Insert (wprowadzanie).

---> Po wciśnięciu o kursor znajdzie się w otwartej linii w trybie Insert.

4. By otworzyć linię POWYŻEJ kursora, wciśnij wielkie O zamiast małego o . Wypróbuj to na linii poniżej.

---> Otwórz linię powyżej wciskając SHIFT-O gdy kursor będzie na tej linii.

~~~~~  
~~~~~

Lekcja 6.2.: POLECENIE

APPEND (dodaj)

** Wpisz a by dodać tekst ZA kursorem. **

1. Przenieś kursor do początku pierwszej linii poniżej oznaczonej --->

2. Wciskaj e dopóki kursor nie będzie na końcu li .

3. Wpisz a (małe), aby dodać tekst ZA znakiem pod kursorem.

4. Dokończ wyraz tak, jak w linii poniżej. Wciśnij <ESC> aby opuścić tryb

Insert.

5. Użyj e by przejść do kolejnego niedokończonego wyrazu i powtarzaj kroki 3. i 4.

---> Ta linia pozwoli Ci ćwiczyć dodawanie tekstu do końca linii.

Uwaga: a , i oraz A prowadzą do trybu Insert, jedyną różnicą jest miejsce, gdzie nowe znaki będą dodawane.

~~~~~  
~~~~~

Lekcja 6.3.: INNA WERSJA
REPLACE (zamiana)

** Wpisz wielkie R by zamienić więcej niż jeden znak. **

1. Przenieś kursor do pierwszej linii poniżej oznaczonej --->. Przenieś kursor do pierwszego xxx .

2. Wciśnij R i wpisz numer poniżej w drugiej linii, tak, że zastąpi on xxx.

3. Wciśnij <ESC> by opuścić tryb Replace. Zauważ, że reszta linii pozostaje niezmienną.

5. Powtarzaj kroki by wymienić wszystkie xxx.

---> Dodanie 123 do xxx daje xxx.

---> Dodanie 123 do 456 daje 579.

UWAGA: Tryb Replace jest jak tryb Insert, ale każdy znak usuwa istniejący znak.

~~~~~  
~~~~~

Lekcja 6.4.: KOPIOWANIE I
WKLEJANIE TEKSTU

** użyj operatora `y` aby skopiować tekst i `p` aby go wkleić **

1. Przejdź do linii oznaczonej `--->` i umieść kursor za "a)".

2. Wejdź w tryb Wizualny `v` i przenieś kursor na początek "pierwszy".

3. Wciśnij `y` aby kopiować (yankować) podświetlony tekst.

4. Przenieś kursor do końca następnej linii: `j$`

5. Wciśnij `p` aby wkleić (wpakować) tekst. Dodaj: `a drugi<ESC>` .

6. Użyj trybu Wizualnego, aby wybrać "element.", yankuj go `y` , przejdź do końca następnej linii `j$` i upakuj tam tekst z `p` .

`--->` a) to jest pierwszy element.

b)

Uwaga: możesz użyć `y` jako operatora; `yw` kopiuje jeden wyraz.

~~~~~  
~~~~~

Lekcja 6.5.:

USTAWIANIE OPCJI

** Ustawianie opcji tak, by szukaj lub substytucja ignorowały wielkość liter **

1. Szukaj 'ignore' wpisując:

`/ignore<ENTER>`

Powtórz szukanie kilka razy naciskając klawisz `n` .

2. Ustaw opcję 'ic' (Ignorecase -- ignoruj wielkość liter) poprzez

wpisanie: :set ic

3. Teraz szukaj 'ignore' ponownie wciskając: n

Zauważ, że Ignore i IGNORE także są teraz znalezione.

4. Ustaw opcje 'hlsearch' i 'incsearch': :set hls is

5. Teraz wprowadź polecenie szukaj ponownie i zobacz co się zdarzy:

/ignore<ENTER>

6. Aby wyłączyć ignorowanie wielkości liter: :set noic

Uwaga: Aby usunąć podświetlanie dopasowań, wpisz: :nohlsearch

Uwaga: Aby ignorować wielkość liter dla jednego wyszukiwania: /ignore\c<ENTER>

~~~~~  
~~~~~

LEKCJA 6.

PODSUMOWANIE

1. Wpisanie o otwiera linię PONIŻEJ kursora.

Wpisanie O otwiera linię POWYŻEJ kursora.

2. Wpisanie a wstawia tekst ZA znakiem, na którym jest kursor.

Wpisanie A dodaje tekst na końcu linii.

3. Polecenie e przenosi do końca wyrazu.

4. Operator y yankuje (kopiuje) tekst, p pakuje (wkleja) go.

5. Wpisanie wielkiego R wprowadza w tryb Replace (zamiana) dopóki

nie zostanie wciśnięty <ESC>.

6. Wpisanie ":set xxx" ustawia opcję

"xxx". Niektóre opcje:
 'ic' 'ignorecase' ignoruj
wielkość znaków
 'is' 'incsearch' pokaż częściowe
dopasowania
 'hls' 'hlsearch' podświetl
wszystkie dopasowania
Możesz użyć zarówno długiej, jak i
krótkiej formy.
7. Dodaj "no", aby wyłączyć opcję: :set
noic

~~~~~  
~~~~~

LEKCJA 7.1. JAK

UZYSKAĆ POMOC?

** Użycie systemu pomocy
on-line **

Vim posiada bardzo dobry system pomocy on-
line. By zacząć, spróbuj jednej
z trzech możliwości:

- wciśnij klawisz <HELP> (jeśli taki masz)
- wciśnij klawisz <F1> (jeśli taki masz)
- wpisz :help<ENTER>

Przeczytaj tekst w oknie pomocy, aby
dowiedzieć się jak działa pomoc.

wpisz CTRL-W CTRL-W aby przeskoczyć z
jednego okna do innego

wpisz :q<ENTER> aby zamknąć okno
pomocy.

Możesz też znaleźć pomoc na każdy temat
podając argument polecenia ":help".

Spróbuj tych (nie zapomnij wcisnąć
<ENTER>):

:help w

```
:helpc_CTRL-D
:help insert-index
:helpuser-manual
```

```
~~~~~
~~~~~
```

LEKCJA 7.2. TWORZENIE SKRYPTU STARTOWEGO

**** Włącz możliwości**

Vima ******

Vim ma o wiele więcej możliwości niż Vi, ale większość z nich jest domyślnie wyłączona. Jeśli chcesz włączyć te możliwości na starcie musisz utworzyć plik "vimrc".

1. Początek edycji pliku "vimrc" zależy od Twojego systemu:

```
:edit ~/.vimrc      dla Uniksa
:edit $VIM/_vimrcdla MS-Windows
```

2. Teraz wczytaj przykładowy plik "vimrc":

```
:read $VIMRUNTIME/vimrc_example.vim
```

3. Zapisz plik:

```
:w
```

Następnym razem, gdy zaczniesz pracę w Vimie będzie on używać podświetlania składni. Możesz dodać wszystkie swoje ulubione ustawienia do tego pliku "vimrc".

Aby uzyskać więcej informacji, wpisz `:helpvimrc-intro`

```
~~~~~
~~~~~
```

Lekcja 7.3.:

UZUPEŁNIANIE

**** Uzupełnianie linii poleceń z CTRL-D i <TAB> ****

1. Upewnij się, że Vim nie jest w trybie kompatybilności: `:set nosp`

2. Zerknij, jakie pliki są w bieżącym katalogu: `!ls` lub `!dir`

3. Wpisz początek polecenia: `:e`

4. Wciśnij `CTRL-D` i Vim pokaże listę poleceń, jakie zaczynają się na "e".

5. Wciśnij `<TAB>` i Vim uzupełni polecenie do `":edit"`.

6. Dodaj spację i zacznij wpisywać nazwę istniejącego pliku: `:edit FIL`

7. Wciśnij `<TAB>`. Vim uzupełni nazwę (jeśli jest niepowtarzalna).

UWAGA: Uzupełnianie działa dla wielu poleceń. Spróbuj wcisnąć `CTRL-D` i `<TAB>`.

Użyteczne zwłaszcza przy `:help` .

~~~~~  
~~~~~

Lekcja 7.

PODSUMOWANIE

1. Wpisz `:help` albo wciśnij `<F1>` lub `<Help>` aby otworzyć okno pomocy.

2. Wpisz `:helpcmd` aby uzyskać pomoc o `cmd` .

3. Wpisz `CTRL-W CTRL-W` aby przeskoczyć do innego okna.

4. Wpisz `:q` aby zamknąć okno pomocy.

5. Utwórz plik startowy `vimrc` aby zachować wybrane ustawienia.

6. Po poleceniu `:` , wciśnij `CTRL-D` aby zobaczyć możliwe uzupełnienia.

Wciśnij `<TAB>` aby użyć jednego z nich.

~~~~~  
~~~~~

Tutaj się kończy tutorial Vima. Został on pomyślany tak, aby dać krótki przegląd jego możliwości, wystarczający byś mógł go używać. Jest on daleki od kompletności, ponieważ Vim ma o wiele, wiele więcej poleceń.

Dla dalszej nauki rekomendujemy książkę:
Vim - Vi Improved - autor Steve Oualline
Wydawca: New Riders
Pierwsza książka całkowicie poświęcona Vimowi. Użyteczna zwłaszcza dla początkujących. Zawiera wiele przykładów i ilustracji.
Zobacz
<http://iccf-holland.org/click5.html>

Starsza pozycja i bardziej o Vi niż o Vimie, ale także warta polecenia:
Learning the Vi Editor - autor Linda Lamb
Wydawca: O'Reilly&Associates Inc.
To dobra książka, by dowiedzieć się niemal wszystkiego, co chciałbyś zrobić z Vi. Szósta edycja zawiera też informacje o Vimie.

Po polsku wydano:
Edytor vi. Leksykon kieszonkowy - autor Arnold Robbins
Wydawca: Helion 2001 (O'Reilly).
ISBN: 83-7197-472-8
<http://helion.pl/ksiazki/vilek.htm>
Jest to książeczka zawierająca spis poleceń vi i jego najważniejszych klonów (między innymi Vima).

Edytor vi - autorzy Linda Lamb i

Arnold Robbins

Wydawca: Helion 2001 (O'Reilly) - wg
6. ang. wydania

ISBN: 83-7197-539-2

<http://helion.pl/ksiazki/viedyt.htm>

Rozszerzona wersja Learning the Vi Editor
w polskim tłumaczeniu.

Ten tutorial został napisany przez
Michaela C. Pierce'a i Roberta K. Ware'a,
Colorado School of Mines korzystając z
pomocy Charlesa Smitha,
Colorado State University.
E-mail: bware@mines.colorado.edu.

Zmodyfikowane dla Vima przez Brama
Moolenaar.

~~~~~  
~~~~~

Przetłumaczone przez Mikołaja
Machowskiego,
Sierpień 2001,
rev. Marzec 2002
2nd rev. Wrzesień 2004
3rd rev. Marzec 2006
4th rev. Grudzień 2008
Wszelkie uwagi proszę kierować na:
mikmach@wp.pl

+++++

W tym tutorialu staram się podać najbardziej podstawowe polecenia używane w VIM'ie, w nagłówkach każdej sekcji w nawiasie podam w jakim trybie trzeba je wpisywać. Tryby te to: NORMAL, INSERT, VISUAL. Z trybu NORMAL do trybu INSERT przełączamy się poprzez i lub inne z wymienionych klawiszy w sekcji Edycja tekstu -> Dodawanie tekstu. Z trybu NORMAL do trybu VISUAL przełączamy się (oraz wyłączamy tryb VISUAL) poprzez klawisz v. Z trybu INSERT/VISUAL do trybu NORMAL przełączamy się za pomocą klawisza ESC.

ZAŁOŻENIA:

- Wiesz co to tryby VIM'a. Jeśli nie, to odsyłam Cię do Wikipedii, albo `manvim` :)
- Wszystkie polecenia zaczynające się od dwukropka (:) zakończone są naciśnięciem klawisza ENTER!
- W nawiasach <> wpisane są nazwy przycisków klawiatury! (np. <ENTER>, <HOME>, itd.)

Wszystkie te polecenia (oraz dużo dużo więcej) możesz znaleźć w dokumentacji, poprostu zebrałem je w jedną całość, w razie gdyby ktoś zapomniał i nie chciał mu się specjalnie szukać :)

1. Wchodzenie do VIM'a (przez konsolę):

- o vim - Otwiera VIM'a (bez skojarzonego pliku). Podczas zapisu trzeba będzie podać nazwę pliku, patrz :w nazwa poniżej.
- o vim nazwa.txt - Otwiera VIM'a wraz ze skojarzonym plikiem (użyteczne podczas pisania kodu np w C, php, html, itp., gdy wpiszymy odrazunazwe pliku i rozszerzenie (np 1.c) oraz gdy opcja kolorowania składni będzie włączona, VIM automatycznie będzie kolorował składnie dla danego typu plików), podczas zapisu wystarczy :w.
- o vim plik1 plik2 - Otwieranie z powłoki systemowej dwóch plików: plik1, plik2.
- o vim *.c - Otwieranie z powłoki systemowej wszystkich plików z rozszerzeniem .c.

2. Wychodzenie z VIM'a (-- NORMAL --):

- o :q - Zamyka plik pod warunkiem, że nie był zmodyfikowany, jeśli był, patrz niżej.
- o :q! - Zamyka plik bez zapisywania zmian.
- o :w - Zapisuje zmiany.
- o :w nazwa - Zapisuje zmiany do pliku o nazwie 'nazwa'.
- o :wq - Zapisuje zmiany i zamyka plik. Gdy edytujemy jeden plik, możemy zapisać i zakończyć używając polecenia :wq i włączyć drugi plik do edycji za pomocą vim plik2, gdzie plik2 to nazwa drugiego pliku. VIM oferuje inne, wygodniejsze rozwiązanie.
- o :vi plik2, gdzie plik2 to nazwa drugiego pliku. Włączamy to z poziomu vima, edytując plik pierwszy. Jeśli nie zapiszemy zmian w pliku pierwszym, VIM nas o tym poinformuje.
- o :vi! plik2, gdzie plik2 to nazwa drugiego pliku. Plik, który aktualnie edytujemy zostanie zamknięty, bez zapisywania ostatnio dokonanych zmian. Oraz zostanie włączony plik2 do edycji.
- o :e plik2 Działa dokładnie tam samo, jak powyższe polecenie :vi plik2
- o :view plik2 Działa prawie tak samo jak polecenia wyżej (:vi plik2, :e plik2) z tym, że plik2 zostanie otwarty tylko do odczytu.

3. Poruszanie się (-- NORMAL --):

- o h / strzałka w lewo (również -- INSERT --) - lewo
- o l / strzałka w prawo (również -- INSERT --) - prawo

- o j / strzałka w dół (również -- INSERT --) - dół
- o k / strzałka do góry (również -- INSERT --) - góra
- o O / <HOME> - Przesuwa kursor na początek
- o \$ / <END> - Przesuwa kursor na koniec
- o <^> - Przesuwa kursor na pierwszy nie-biały znak, np w zdaniu " Ala ma kota", kursor zostanie ustawiony na literę 'A'
- o w - Przesuwa kursor na początek następnego wyrazu
- o e - Przesuwa kursor na koniec następnego wyrazu
- o b - Przesuwa kursor na początek poprzedniego wyrazu
- o ge - Przesuwa kursor na koniec poprzedniego wyrazu
- o gg - Przesuwa kursor do pierwszego wiersza w pliku
- o G - Przesuwa kursor do ostatniego wiersza w pliku
- o :n - Przesuwa kursor do wiersza n - tego (:12 - Przesuwa kursor do wiersza 12).
- o nG - gdzie n to numer wiersza. Działanie takie samo jak :n (np 30G - przeniesie do 30 wiersza)
- o :\$ - To samo co G.
- o CTRL-G - Pokazuje położenie w pliku i status pliku
- o CTRL-U - Przesuwa ekran o połowę tekstu do góry.
- o CTRL-D - Przesuwa ekran o połowę tekstu w dół.
- o :next - Przechodzi do następnego pliku, pod warunkiem, że plik aktualny nie był zmodyfikowany, jeśli był, VIM pokaże ostrzeżenie.
- o :previous - Analogicznie do polecenia wyżej, tylko, że przechodzi do poprzedniego pliku.
- o :wnext - Przechodzi do następnego pliku, oraz zapisuje aktualnie edytowany plik.
- o :wprevious - Analogicznie do polecenie wyżej, tylko, że przechodzi do poprzedniego pliku.
- o :next! - Przechodzi do następnego pliku, nie zapisując zmian jakie zostały zrobione w aktualnym pliku.
- o :previous! - Analogicznie do polecenia wyżej, tylko, że przechodzi do poprzedniego pliku.
- o :2next - Przesuwa się nie na następny a jeden dalej plik. Analogicznie istnieją opcję w tył, w przód z/bez zapisu.

- o :args - Pokazuje otwarte pliki, oraz w nawiasach kwadratowych, aktualnie edytowany plik, np. plik1.txt [plik2.txt] plik3.txt
- o :rewind - Przenosi do pierwszego otwartego pliku (zakładamy, że mamy otwartych kilka plików) nie ważne, który aktualnie plik edytujemy.
- o :first - To samo co wyżej.
- o :last - Przenosi do ostatniego otwartego pliku (zakładamy, że mamy otwartych kilka plików) nie ważne, który aktualnie plik edytujemy.

4. Mnożniki dla ruchu (-- NORMAL --):

- o Można zastosować mnożnik ruchu, który pozwoli przesunąć kursor o większą ilość wyrazów, np.
- o 2w - Przesuwa kursor na początek 2 wyrazu w prawo (licząc od pozycji kursora).
- o 2e - Przesuwa kursor na koniec 2 wyrazu w prawo (licząc od pozycji kursora).
- o 2b - Przesuwa kursor na początek 2 wyrazu w lewo (licząc od pozycji kursora).
- o 2ge - Przesuwa kursor na koniec 2 wyrazu w lewo (licząc od pozycji kursora). itp, itd.
- o 9k - Przesuwa kursor o 9 pozycji w górę (równoważne z: kkkkkkkkk)
- o 3j - Przesuwa kursor o 3 pozycje w dół (równoważne z: jjj)
- o 5l - Przesuwa kursor o 5 pozycji w prawo (równoważne z: lllll)
- o 2h - Przesuwa kursor o 2 pozycje w lewo (równoważne z: hh)

5. Edycja tekstu (-- NORMAL --):

- o Dodawanie tekstu:(-- NORMAL --)
- ♣ i - Wstawianie tekstu przed kursorem.
- ♣ a - Wstawianie tekstu za kursorem.
- ♣ A - Wstawianie tekstu na końcu linii.
- ♣ o - Wstawianie tekstu do nowej (następnej) linii.
- ♣ O - Wstawianie tekstu do nowej (poprzedniej) linii.

o Mnożniki używane podczas wstawiania tekstu są bardzo użyteczne. Np, jeśli chciałbyś wstawić 10 liter 'A' obok siebie, lub każda w nowym wierszu, oczywiście możesz to wpisywać ręcznie, ale VIM udostępnia mnożniki, tak więc, aby wpisać 10 liter 'A' szybciej, przejdź do trybu NORMAL oraz:

♣ 10iA<ESC> - Wstawienie 10 liter 'A' za kursorem

♣ 10oFraza<ESC> - Wstawienie 10 wyrazów 'Fraza', każdy w nowej linii. Zauważ, że po naciśnięciu '10o' VIM przechodzi w tryb INSERT, co za tym idzie, wszystko co wpiszesz, po naciśnięciu ESC zostanie skopiowane tyle razy ile podałeś w poleceniu (tutaj 10 razy).

o Kopiowanie i wklejanie tekstu (-- NORMAL -- / -- VISUAL --):

♣ y - Operator kopiowania, używać go można w trybie wizualnym (opisany na dole), lub z operatorami ruchu.

♣ yw - Kopiuje jeden wyraz

♣ y3w - Kopiuje 3 wyrazy

♣ :ny - Gdzie n to numer wiersza, np. :30y - Skopiuje wiersz 30.

♣ p - Operator wklejania.

♣ np - Gdzie n to wielokrotność kopiowanego tekstu, np. 10p - Wkleji 10 razy, to co zostało skopiowane.

o Łączenie linii (-- NORMAL --):

♣ J - Łaczy 2 linie (kursor trzeba ustawić na najwyższej). Pomiedzy liniami dodaje spacje

♣ 5J - Łaczy 5 linii, między każdą dodaje spacje.

o Usuwanie tekstu (-- NORMAL --):

♣ x / del - Usuwa znak pod kursorem.

♣ dw - Usunięcie jednego wyrazu do pierwszego znaku nowego wyrazu (innymi słowami: usuwa wyraz i spacje przed następnym wyrazem).

♣ de - Usunięcie jednego wyrazu do ostatniego znaku usuwanego wyrazu (innymi słowami: usuwa tylko wyraz a spacja przed następnym wyrazem zostaje).

♣ d\$ - Usunięcie tekstu od kursora do końca linii.

♣ dd - Usunięcie całego wiersza niezależnie od położenia kursora.

- o Mnożniki: (-- NORMAL --)
- ♣ Mnożniki dla edycji tekstu działają właściwie tak samo, jak mnożniki ruchu.
- ♣ d2e - Usunięcie 2 wyrazów.
- ♣ d2w - Usunięcie 2 wyrazów (różnice pomiędzy de i dw zostały opisane wyżej).
- ♣ d3d - Usunięcie 3 linii, itp. itd.
- o Oznaczenie (Mark) (-- NORMAL --):
- ♣ ma - 'm' to operator zaznaczenia, 'a' to tak jakby miejsce zaznaczenia. Można wybrać zamiast 'a' litery od a do z. Operator zaznaczenia przydaje się np do usuwania znaczniej ilości bloków tekstu.
- ♣ 'a - Po tym jak zaznaczymy pewne miejsce przez operator 'm', to odwołujemy się do niego poprzez 'a (apostrof a, jak wyżej wspomniano, zamiast a, może być inna litera alfabetu).
- ♣ d'a - Dzięki takiej konstrukcji, VIM usuwa wszystko od miejsca zaznaczenia (ma) do miejsca, w którym znajduje się kursor
- ♣ y'a - Dzięki takiej konstrukcji, VIM kopiuje wszystko od miejsca zaznaczenia (ma) do miejsca, w którym znajduje się kursor
- ♣ :marks - Pokazuje listę wszystkich zaznaczeń.
- ♣ :marksargs - Zamiast 'args' możesz wpisać konkretne zaznaczenie, np. a, b, itp., które chcesz zobaczyć. Lub wpisać to polecenie tak jak jest, razem z args.
- o Cofanie (-- NORMAL --):
- ♣ u - Cofa skutki ostatniego polecenia
- ♣ U - Cofa skutki dla całej linii
- ♣ CTRL-R - Cofa cofnięcia, innymi słowy: Cofnij/Do przodu.
- o Wstawianie (-- NORMAL --):
- ♣ p - Wstawia ostatnio usunięte / skopiowane znaki.
- o Zamienianie (-- NORMAL --):
- ♣ r - Po naciśnięciu tego klawisza VIM czeka, aż podasz znak, na który ma zamienić wskazywany przez kursor znak.
- ♣ 5ra - Zamienia 5 znaków na prawo od aktualnej pozycji kursora na literę 'a'.
- ♣ R - Zamienia więcej niż jeden znak, działa na takiej samej zasadzie, jakbyś pisał z wciśniętym insertem (przycisk na klawiaturze, nie tryb pracy VIM'a).

- ♣ ce - Usuwa wyraz od miejsca wskazywanego przez kursor do końca wyrazu, przechodzi w tryb INSERT i czeka na wprowadzenie wyrazu.
- ♣ cw - Prawie to samo co ce, różnica jest taka sama jak pomiędzy de i dw.
- ♣ c\$ - Usuwa linie od miejsca wskazywanego przez kursor do końca linii, przechodzi w tryb INSERT i czeka na wprowadzenie wyrazu.
- ♣ Mnożniki również tu występują. Zasada jest analogiczna: cnw, gdzie n to liczba całkowita (1, 2, ..., n), zamiast w, może być e, \$, itp.
- ♣ c2e - Usuwa 2 wyrazy od miejsca wskazywanego przez kursor do końca 2 wyrazu i przechodzi w tryb INSERT, czeka na wprowadzenie nowych wyrazów.
- ♣ ~ - (tylda) Zamienia znak nad którym znajduje się kursor z dużego na mały i odwrotnie.
- ♣ 10~ - Zmienia wielkość 10 znaków na prawo od miejsca, w który znajduje się kursor.
- ♣ xp - Przydatne w sytuacji, gdy popełnisz literówkę, np. 'teh' zamiast 'the', ustaw kursor w słowie 'teh' na 'e' i wydaje to polecenie.
- o Szukaj (-- NORMAL --):
- ♣ /nazwa_blad<ENTER> - VIM szuka od początku pliku frazy: nazwa_blad
- ♣ ?nazwa_blad<ENTER> - VIM szuka od tyłu pliku frazy: nazwa_blad
- ♣ /<ENTER> - Znajduje kolejne wystąpienie słowa szukanego, w tym przypadku będzie to nazwa_blad
- ♣ /^word<ENTER> - Znajduje wyraz 'word' jeśli on zaczyna linię.
- ♣ /word\$<ENTER> - Znajduje wyraz 'word' jeśli on kończy linię.
- ♣ /c.m<ENTER> - Znajduje słowo zaczynające się na c, drugi znak dowolny a trzeci znak to m.
- ♣ /xxx\c<ENTER> - Wyszukuje frazę 'xxx' bez uwzględnienia wielkości liter (przełącznik \c).
- ♣ n - VIM szuka kolejnych wystąpień szukanej frazy od początku do końca.
- ♣ 10n - Szuka dziesiątego wystąpienia szukanej frazy w przód.
- ♣ N - VIM szuka kolejnych wystąpień szukanej frazy od końca do początku.
- ♣ 10N - Szuka dziesiątego wystąpienia szukanej frazy w tył
- ♣ /<UP> - <UP> - Strzałka do góry. Przeszukuje historie szukania.
- ♣ /<DOWN> - <DOWN> - Strzałka w dół. Przeszukuje historie szukania.
- ♣ CTRL-O - Przenosi kursor w poprzednie wystąpienie szukanej frazy (tam, gdzie byłeś)
- ♣ CTRL-I - Przenosi kursor w następne wystąpienie szukanej frazy (tam, gdzie byłeś)

- ♣ fx - Szuka znaku 'x' na prawo od kursora. Ustawia kursor na pierwszym znalezionym znaku 'x'.
- ♣ 5fa - Przeszukiwanie oraz ustawienie kursora na piątej literze 'a' występującej na prawo od miejsca kursora
- ♣ 10f(spacja) - Przeszukiwanie oraz ustawienie kursora na 10 spacji występującej na prawo od miejsca kursora
- ♣ Fi - Szuka znaku 'i' na lewo od kursora. Ustawia kursor na pierwszym znalezionym znaku 'i'
- ♣ 2Fo - Przeszukiwanie oraz ustawianie kursora na drugiej literze 'o' występującej na lewo od miejsca kursora
- ♣ to - Szuka znaku 'o' na prawo od kursora. Ustawia kursor na znaku poprzedzającym znak 'o'.
- ♣ To - Szuka znaku 'o' na lewo od kursora. Ustawia kursor na znaku poprzedzającym znak 'o'.
- o Szukanie brakujących nawiasów ((,), [,], {, })(-- NORMAL --):
- ♣ % - Szuka brakującego nawiasu, przenosi do miejsca, w którym występuje zamykający/otwierający brakujący nawias.
- o Substitute (znajdź i zamień)(-- NORMAL --):
- ♣ :s/xxx/yyy - Zamienia frazę 'xxx' na 'yyy' w linii, gdzie znajduje się kursor. Zauważ, że polecenie to zamieni tylko jedno wystąpienie frazy 'xxx', co jeśli chcemy zamienić wszystkie wystąpienia tejże frazy w tej linii? odpowiedź niżej.
- ♣ :s/xxx/yyy/g - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' w linii, w której znajduje się kursor.
- ♣ :s/xxx/yyy/gc - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' w lini, w której znajduje się kursor, prosząc za każdym razem o potwierdzenie.
- ♣ :10s/xxx/yyy - Zamienia frazę 'xxx' na 'yyy' w linii nr. 10 (kursor nie musi być ustawiony w tej linii, polecenie to jest bardziej precyzyjne)
- ♣ :10s/xxx/yyy/g - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' w linii nr. 10
- ♣ :10s/xxx/yyy/gc - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' w linii nr. 10, za każdym razem prosząc o potwierdzenie akcji.
- ♣ :%s/xxx/yyy/g - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' w całym pliku
- ♣ :%s/xxx/yyy/gc - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' w całym pliku, lecz za każdym razem prosząc o potwierdzenie.
- ♣ :#,s/xxx/yyy/g - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' zawierającej się pomiędzy wierszami #, #.
- ♣ :#,s/xxx/yyy/gc - Zamienia wszystkie wystąpienia frazy 'xxx' na 'yyy' zawierającej się pomiędzy wierszami #,# za każdym razem prosząc o potwierdzenie

o Polecenie zewnętrzne (-- NORMAL --):

♣ `!:nazwa_polecenia` - powłoka wykona polecenie 'nazwa_polecenia' (np. `!:ls -l` pokaże listę plików katalogu, w którym jesteśmy).

o Wybranie tekstu do zapisu (-- VISUAL --):

♣ `v` - Przełącza do trybu wizualnego. Możesz używać strzałek do zaznaczenia tekstu i jak już pewnie zobaczyłeś tekst jest podświetlony, ale również możesz używać operatorów ruchu: `w` (słowo), `e` (słowo), `$` (cała linia od kursora). Wciśnij `:`, a na dole pojawi się coś takiego: `:'<,>`. Wpisz za tym `'w nazwa'` (oczywiście bez `"`), gdzie nazwa to nieistniejącego pliku. Mniej więcej powinno wyglądać to tak: `:'<,>w nazwa`. Teraz, aby sprawdzić, czy zaznaczony tekst zapisał się do nowego pliku wystarczy wpisać `!:cat nazwa`

o Filtrowanie (-- NORMAL --):

♣ `!20Gsort<ENTER>` - Jeśli ustawimy 20 wierszy, w każdym będzie jakaś liczba, to po wydaniu tego polecenia, VIM posortuje je rosnąco.

♣ `!Gsort<ENTER>` - Posortuje wiersze od miejsca, w którym znajduje się kursor do końca pliku.

♣ `!!ls -l` - Wstawi listę znajdujących się plików w katalogu. Dla użytkowników MS Windows: `!!dir`

♣ `!!date` - Wstawi datę systemową.

o Wstawienie i łączenie plików (-- NORMAL --):

♣ `:r nazwa` - Gdzie nazwa to nazwa pliku wstawianego. Można również wstawić zawartość polecenia zewnętrznego, tj. `:r !ls -l`

6. Ustawianie opcji (-- NORMAL --):

o `:set ic` - `ic` (Ignore Case) po ustawieniu tej opcji, wyrazy będą przeszukiwane bez względu na wielkość ich liter.

o `:set noic` - Wyłącza opcję `ic`.

o `:set hlsearch` - Włącza podświetlenie wyszukiwanego tekstu.

o `:set nohl` - Wyłącza podświetlenie wyszukiwanego tekstu.

o `:set incsearch` - Włącza częściowe dopasowanie tekstu podczas szukania.

o `:set noxxx` - Wyłącza opcję `xxx`

o `:set autowrite` - Włącza opcję autozapisywania.

o `:set noautowrite` - Wyłącza opcję autozapisywania.

7. Używanie tabulacji (-- NORMAL --)

- o << - (nawias trójkątny) Przesuwa linię w lewo, w której znajduje się kursor o wartość tabulacji.
- o >> - Przesuwa linię w prawo, w której znajduje się kursor o wartość tabulacji.
- o #,> - #,# - Numery wierszy. Przesuwa w prawo o wartość tabulacji wiersze podane jako argumenty.
- o #,<< - #,# - Numery wierszy. Przesuwa w lewo o 2 wartości tabulacji wiersze podane jako argumenty.

8. Używanie Makr (-- NORMAL --)

Posiadasz taką o to listę:

- o stdio.h
- o stdlib.h
- o sys/types.h
- o sys/stat.h
- o unistd.h
- o fcntl.h
- o errno.h

I chcesz, żeby wyglądała ona tak (za pomocą jednego polecenia):

- o #include <stdio.h>
- o #include <stdlib.h>
- o #include <sys/types.h>
- o #include <sys/stat.h>
- o #include <unistd.h>
- o #include <fcntl.h>
- o #include<errno.h>

Wykonuj pokolei wypunktowane poniżej polecenia, ale zanim to zrobisz przenieś kursor do pierwszego znaku w pierwszej linii:

- o qa - Zaczyna nagrywać makro, przypisując je literze a. Litera musi być z przedziału <a,z>
- o ^ - Przenosi na początek linii

- o `i#include<<ESC>` - Wstawia napis `#include<` na początku linii. (pamiętaj, że to `<ESC>` to jest przycisk na klawiaturze!)
- o `$` - Przechodzi na koniec linii
- o `a><ESC>` - Dodaje znak: `>` na końcu linii.
- o `j` - Przechodzi do następnej linii
- o `q` - Kończy nagrywanie makra

Teraz, żeby się odwołać do tego makra, przenieś kursor na pierwszy znak w pierwszej linii i w trybie normal wydaj polecenie: `@a`. Żeby to zrobić dla wszystkich naraz wpisz: `7@a`.

9. Dwuznaki (Digraphs) (-- NORMAL -- / -- INSERT --):

- o `:digraphs` - Wyświetla listę dostępnych znaków
- o `CTRL-Kxy` - (Tryb INSERT), x, y - oznaczają znaki, które można odczytać z `:digraphs`. Dajmy na to przykład, żeby otrzymać znak: `Ł` (`C= Ł 1062`, Informacja z `:digraphs`) musimy nacisnąć `CTRL-K` i wpisać `C=` (dwa znaki: `C, =`)
- o `CTRL-KD%` - Analogicznie da inny znak, w tym wypadku ten znak to: `Ѓ`. (Informacje z `digraphs: D% Ѓ 1026`)

10. Okna (-- NORMAL --)

- o `:split` - Polecenie to dzieli okno VIMa na 2 okna, w których edytuje ten sam plik.
- o `CTRL-Ww` / `CTRL-W CTRL-W` - Przełącza okna
- o `CTRL-Wj` - Przełącza z górnego okienka do dolnego
- o `CTRL-Wk` - Przełącza z dolnego okienka do górnego
- o `ZZ` / `:q` / `CTRL-Wc` - Zamyka jedno okienko
- o `:split plik2.txt` - Otwiera w drugim okienki plik: `plik2.txt`
- o `:15 split plik2.txt` - Otwiera `plik2.txt` w drugim oknie o rozmiarze 15 wierszy
- o `:new` - Prawie to samo co `split`, tylko, że otwiera nowy plik, w oknie.
- o `:10 new` - Otwiera nowy plik w oknie o rozmiarze 10 wierszy.
- o `:sview plik2.txt` - Otwiera `plik2.txt` w nowym oknie, tylko do odczytu.
- o `10 sview plik2.txt` - Otwiera `plik2.txt` w nowym oknie o rozmiarze 10 wierszy, tylko do odczytu.
- o `10CTRL-W+` - Powiększa okno o 10 wierszy.
- o `10CTRL-W-` - Pomniejsza okno o 10 wierszy.

- o 12CTRL-W_ - Ustawia rozmiar okna na 12 wierszy.
- o CTRL-W_ - Ustawia okno na największą możliwość
- o CTRL-W= - Ustawia okna na takie same rozmiary (albo bardzo zbliżone)