Command Action

H or backspace Moves Cursor Left

l or spacebar Moves Cursor Right

j Moves Cursor Down

K Moves Cursor Up

nG Go to line number n

^ / 0 / | Moves cursor to beginning of line

$ Moves cursor to end of line

w Moves cursor to next word

B Moves cursor back to previous word

e Moves cursor forward to end of word

Note: W, B, E perform functions similar to w, b, e except that punctuation is

skipped)

Command Action

ZZ saves file only if vi started with a filename

Save and Exit Commands of ex Mode

:w save file and remain in edit mode

:wq save file and quit edit mode

:w <filename> write buffer to filename

:q quit editing mode when no changes are made

:q! quit editing mode but after abandoning changes

:x save file and quit editing mode

Command Action

a Appends text to right of cursor

A Appends text at end of line

i Inserts text to left of cursor

I Inserts text at beginning of line

/ Inserts before current character

o Inserts blank line below + inserts text

O Inserts blank line above + inserts text

rx Replace current character with char x

Rtext<Esc> Replaces ccharacterharacter with text

s Replaces character under cursor withmany characters

S Replaces entire line

20i- followed by <Enter> or <Esc> enters

20 hyphens (-) in one line

//windows ka code bunk

while [ condition ]

do

command1

command2

commandN

done

ksh while loop syntax

while [[ condition ]] ; do

command1

command1

commandN

done

csh while loop syntax

while ( condition )

commands

end

#!/bin/bash

c=1

while [ $c -le 5 ]

do

echo "Welcone $c times"

(( c++ ))

done

KSH while loop Example

#!/bin/ksh

c=1

while [[ $c -le 5 ]]; do

echo "Welcome $c times"

(( c++ ))

done

CSH while loop Example

#!/bin/csh

c=1

while ( $c <= 5 )

echo "Welcome $c times"

@ c = $c + 1

end

Another example:

#!/bin/csh

set yname="foo"

while ( $yname != "" )

echo -n "Enter your name : "

set yname = $<

if ( $yname != "" ) then

echo "Hi, $yname"

endif

end

// linux ka codebunk

file=testfile.txt

while read line

do

if [ $line = done ]

then

break

fi

echo $line >> $file

done

file=testfile.txt

while read -r line

do

echo $line

done < "$file"

while :

do

read -p "Enter the number" a b

echo $a

echo $b

if [ $a -eq 1 ]

then

break

fi

ans=$((a+b))

echo "Value of answers: $ans"

done

a=0

while [ $a -lt 10 ]

do

echo $a

#a=`expr $a + 1`

a=$(( $a + 1 ))

done

file=testfile.txt

while read -r line

do

echo $line

done < "$file"

while :

do

read -p "Enter the number" a b

echo $a

echo $b

if [ $a -eq 1 ]

then

break

fi

ans=$((a+b))

echo "Value of answers: $ans"

done

a=0

while [ $a -lt 10 ]

do

echo $a

#a=`expr $a + 1`

a=$(( $a + 1 ))

done

while :

do

read -p "Enter the number" a

if [ $a -eq 1 ]

then

break

fi

ans=$((a+b))

echo "Value of answers: $ans"

done

a=0

while [ $a -lt 10 ]

do

echo $a

#a=`expr $a + 1`

a=$(( $a + 1 ))

done

a=0

while [ $a -lt 10 ]

do

echo $a

a=`expr $a + 1`

done

NOW=$(date +"%a")

echo $NOW

case $NOW in

Mon)

echo "Full Backup"

;;

Tue|Wed|Thu|Fri)

echo "Partial Backup"

;;

Sat|Sun)

echo "No Backup"

;;

\*)

;;

esac

echo "Enter a number"

read num

case $num in

[0-9])

echo "YOu have entered single digit number"

;;

[0-9][0-9])

echo "You have entered double digit number"

;;

[0-9][0-9][0-9])

echo "You have entered three digit number"

;;

\*)

echo "More than 3 digit number"

;;

esac

echo "Enter state name"

read state

case $state in

"Maharashtra")

echo "Capital is Mumbai"

;;

"Uttar Pradesh")

echo "Capital is Lacknow"

;;

"Madhya Pradesh")

echo "Capital is Bhopal"

;;

\*)

echo "Cant find capital city"

;;

esac

#!/bin/bash

#echo "$\*"

options="$1"

echo $options

case ${options} in

-f) FILE="${2}"

echo "File name is $FILE"

;;

-d) DIR="${2}"

echo "Dir name is $DIR"

;;

\*) echo "name ${0} usage [-f file] [-d directory] "

;;

esac

#!/bin/bash

echo "Enter a number"

read n

if [ `expr $n % 2` -eq 0 ]

then

echo "Even"

else

echo "Odd"

fi

#!/bin/bash

a=10

b=20

if [ $a -eq $b ]

then

echo "Both are same"

fi

if [ $a -eq $b ]

then

echo "Both are equal"

elif [ $a -gt $b ]

then

echo "$a is greater than $b"

elif [ $a -lt $b ]

then

echo "$a is less than $b"

else

echo "Condition not matches"

fi

echo "File Name :- $0"

echo "First parameter $1"

echo "Second parameter $2"

echo "Third parameter $3"

echo "Fourth parameter $4"

echo "Fifth parameter $5"

echo "All Parameters are $\*"

echo "Total number of arguments $#"

echo "Process id $$"

echo "All Parameters same like $\* but with () $@"

#!/bin/bash

echo "using \"\$\*\":"

for a in "$\*"; do

echo $a

done

echo "\nUsing \$\*:"

for a in $\*; do

echo $a;

done

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

echo "using \"\$@\":"

for a in "$@"; do

echo $a

done

echo "\nUsing \$@:"

for a in $@; do

echo $a;

done

#for var in "$@"

#do

# echo "$var"

#done

//password manually

var1="welcome"

var2="welcome11"

if [ "$var1" = "$var2" ];then

echo "Strings are equal"

else

echo "String are not equal"

fi

//password with user input comparison

read -p "Enter first string " str1

read -p "Enter second string" str2

#[ "$str1" = "$str2" ] && echo "Equal" || echo "Not equal"

if [ "$str1" = "$str2" ]; then

echo "Strings are equal"

else

echo "Strings are not equal"

fi

var1="welcome"

var2="welcome11"

if [ "$var1" = "$var2" ];then

echo "Strings are equal"

else

echo "String are not equal"

fi

//codebunk linux

read -p "Enter first string " str1

read -p "Enter second string " str2

if [ "$str1" > "$str2" ]; then

echo "${str1} is lexicographically greater than ${str2}"

elif [ "$str1" < "$str2" ]; then

echo "${str1} is lexicographically less than ${str2}"

else

echo "String are equal"

fi

VAR=''

if [ -z $VAR ]; then

echo "String is empty"

fi

MSG='I know linux programming'

if [ "$MSG" = \*"know"\* ]; then

echo "YES I FOUND Linux"

else

echo "Not found"

fi

read -p "Enter first string " str1

read -p "Enter second string" str2

[ "$str1" = "$str2" ] && echo "Equal" || echo "Not equal"

if [ "$str1" = "$str2" ]; then

echo "Strings are equal"

else

echo "Strings are not equal"

fi

read -p "Enter first string " str1

read -p "Enter second string" str2

[ "$str1" = "$str2" ] && echo "Equal" || echo "Not equal"

if [ "$str1" = "$str2" ]; then

echo "Strings are equal"

else

echo "Strings are not equal"

fi

var1="welcome"

var2="welcome11"

if [ "$var1" = "$var2" ];then

echo "Strings are equal"

else

echo "String are not equal"

fi

read -p "Enter first string " str1

read -p "Enter second string " str2

if [ "$str1" > "$str2" ]; then

echo "${str1} is lexicographically greater than ${str2}"

elif [ "$str1" < "$str2" ]; then

echo "${str1} is lexicographically less than ${str2}"

else

echo "String are equal"

fi

read -p "Enter first string " str1

read -p "Enter second string " str2

if [ "$str1" > "$str2" ]; then

echo "${str1} is lexicographically greater than ${str2}"

elif [ "$str1" < "$str2" ]; then

echo "${str1} is lexicographically less than ${str2}"

else

echo "String are equal"

fi