Welcome to the VIM Tutor - Version 1.7

Vim is a very powerful editor that has many commands, too many to explain in a tutor such as this. This tutor is designed to describe enough of the commands that you will be able to easily use Vim as an all-purpose editor.

The approximate time required to complete the tutor is 25-30 minutes, depending upon how much time is spent with experimentation.

ATTENTION:
The commands in the lessons will modify the text. Make a copy of this file to practise on (if you started "vimtutor" this is already a copy).

It is important to remember that this tutor is set up to teach by use. That means that you need to execute the commands to learn them properly. If you only read the text, you will forget the commands!

Now, make sure that your Shift-Lock key is NOT depressed and press the j key enough times to move the cursor so that Lesson 1.1 completely fills the screen.

Lesson 1.1: MOVING THE CURSOR

** To move the cursor, press the h,j,k,l keys as indicated. **

```
             ^
             k           Hint:  The h key is at the left and moves left.
       < h       l >            The l key is at the right and moves right.
             j                  The j key looks like a down arrow.
             v
```

1. Move the cursor around the screen until you are comfortable.

2. Hold down the down key (j) until it repeats.
   Now you know how to move to the next lesson.

3. Using the down key, move to Lesson 1.2.

NOTE: If you are ever unsure about something you typed, press <ESC> to place you in Normal mode. Then retype the command you wanted.

NOTE: The cursor keys should also work. But using hjkl you will be able to move around much faster, once you get used to it. Really!

Lesson 1.2: EXITING VIM

!! NOTE: Before executing any of the steps below, read this entire lesson!!

1. Press the <ESC> key (to make sure you are in Normal mode).

2. Type: `:q! <ENTER>`.
   This exits the editor, DISCARDING any changes you have made.

3. When you see the shell prompt, type the command that got you into this tutor. That would be: `vimtutor <ENTER>`

4. If you have these steps memorized and are confident, execute steps 1 through 3 to exit and re-enter the editor.

NOTE: `:q! <ENTER>` discards any changes you made. In a few lessons you will learn how to save the changes to a file.

```
  5. Move the cursor down to Lesson 1.3.


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                  Lesson 1.3: TEXT EDITING - DELETION


          ** Press  x  to delete the character under the cursor. **

  1. Move the cursor to the line below marked --->.

  2. To fix the errors, move the cursor until it is on top of the
     character to be deleted.

  3. Press the      x  key to delete the unwanted character.

  4. Repeat steps 2 through 4 until the sentence is correct.

---> The ccow jumpedd ovverr thhe mooon.

  5. Now that the line is correct, go on to Lesson 1.4.

NOTE: As you go through this tutor, do not try to memorize, learn by usage.



~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                   Lesson 1.4: TEXT EDITING - INSERTION


                    ** Press  i  to insert text. **

  1. Move the cursor to the first line below marked --->.

  2. To make the first line the same as the second, move the cursor on top
     of the first character AFTER where the text is to be inserted.

  3. Press  i  and type in the necessary additions.

  4. As each error is fixed press <ESC> to return to Normal mode.
     Repeat steps 2 through 4 to correct the sentence.

---> There is text misng this .
---> There is some text missing from this line.

  5. When you are comfortable inserting text move to lesson 1.5.



~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                  Lesson 1.5: TEXT EDITING - APPENDING


                    ** Press  A  to append text. **

  1. Move the cursor to the first line below marked --->.
     It does not matter on what character the cursor is in that line.

  2. Press  A  and type in the necessary additions.

  3. As the text has been appended press <ESC> to return to Normal mode.

  4. Move the cursor to the second line marked ---> and repeat
     steps 2 and 3 to correct this sentence.
```

```
---> There is some text missing from th
     There is some text missing from this line.
---> There is also some text miss
     There is also some text missing here.

  5. When you are comfortable appending text move to lesson 1.6.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                  Lesson 1.6: EDITING A FILE

                 ** Use  :wq  to save a file and exit. **

  !! NOTE: Before executing any of the steps below, read this entire lesson!!

  1. Exit this tutor as you did in lesson 1.2:  :q!
     Or, if you have access to another terminal, do the following there.

  2. At the shell prompt type this command:  vim tutor <ENTER>
     'vim' is the command to start the Vim editor, 'tutor' is the name of the
     file you wish to edit.  Use a file that may be changed.

  3. Insert and delete text as you learned in the previous lessons.

  4. Save the file with changes and exit Vim with:  :wq  <ENTER>

  5. If you have quit vimtutor in step 1 restart the vimtutor and move down to
     the following summary.

  6. After reading the above steps and understanding them: do it.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                           Lesson 1 SUMMARY


  1. The cursor is moved using either the arrow keys or the hjkl keys.
        h (left)    j (down)       k (up)          l (right)

  2. To start Vim from the shell prompt type:  vim FILENAME <ENTER>

  3. To exit Vim type:        <ESC>   :q!       <ENTER>  to trash all changes.
            OR type:   <ESC>   :wq       <ENTER>  to save the changes.

  4. To delete the character at the cursor type:  x

  5. To insert or append text type:
        i   type inserted text   <ESC>         insert before the cursor
        A   type appended text   <ESC>         append after the line

NOTE: Pressing <ESC> will place you in Normal mode or will cancel
      an unwanted and partially completed command.

Now continue with Lesson 2.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                    Lesson 2.1: DELETION COMMANDS


                    ** Type  dw  to delete a word. **

  1. Press  <ESC>  to make sure you are in Normal mode.

  2. Move the cursor to the line below marked --->.

  3. Move the cursor to the beginning of a word that needs to be deleted.
```

4. Type   dw   to make the word disappear.

NOTE: The letter d will appear on the last line of the screen as you type it. Vim is waiting for you to type w . If you see another character than d you typed something wrong; press <ESC> and start over.

---> There are a some words fun that don't belong paper in this sentence.

5. Repeat steps 3 and 4 until the sentence is correct and go to Lesson 2.2.

Lesson 2.2: MORE DELETION COMMANDS

** Type d$ to delete to the end of the line. **

1. Press <ESC> to make sure you are in Normal mode.

2. Move the cursor to the line below marked --->.

3. Move the cursor to the end of the correct line (AFTER the first .).

4. Type   d$ to delete to the end of the line.

---> Somebody typed the end of this line twice. end of this line twice.

5. Move on to Lesson 2.3 to understand what is happening.

Lesson 2.3: ON OPERATORS AND MOTIONS

Many commands that change text are made from an operator and a motion.
The format for a delete command with the d delete operator is as follows:

```
     d   motion
```

Where:
```
  d      - is the delete operator.
  motion - is what the operator will operate on (listed below).
```

A short list of motions:
```
  w - until the start of the next word, EXCLUDING its first character.
  e - to the end of the current word, INCLUDING the last character.
  $ - to the end of the line, INCLUDING the last character.
```

Thus typing de will delete from the cursor to the end of the word.

NOTE: Pressing just the motion while in Normal mode without an operator will move the cursor as specified.

Lesson 2.4: USING A COUNT FOR A MOTION

** Typing a number before a motion repeats it that many times. **

1. Move the cursor to the start of the line marked ---> below.

2. Type 2w to move the cursor two words forward.

3. Type 3e to move the cursor to the end of the third word forward.

4. Type 0 (zero) to move to the start of the line.

5. Repeat steps 2 and 3 with different numbers.

```
---> This is just a line with words you can move around in.
```

6. Move on to Lesson 2.5.

Lesson 2.5: USING A COUNT TO DELETE MORE

** Typing a number with an operator repeats it that many times. **

In the combination of the delete operator and a motion mentioned above you insert a count before the motion to delete more:

```
     d   number   motion
```

1. Move the cursor to the first UPPER CASE word in the line marked --->.

2. Type d2w to delete the two UPPER CASE words

3. Repeat steps 1 and 2 with a different count to delete the consecutive UPPER CASE words with one command

```
--->  this ABC DE line FGHI JK LMN OP of words is Q RS TUV cleaned up.
```

Lesson 2.6: OPERATING ON LINES

** Type dd   to delete a whole line. **

Due to the frequency of whole line deletion, the designers of Vi decided it would be easier to simply type two d's to delete a line.

1. Move the cursor to the second line in the phrase below.
2. Type dd to delete the line.
3. Now move to the fourth line.
4. Type   2dd   to delete two lines.

```
--->  1)  Roses are red,
--->  2)  Mud is fun,
--->  3)  Violets are blue,
--->  4)  I have a car,
--->  5)  Clocks tell time,
--->  6)  Sugar is sweet
--->  7)  And so are you.
```

Lesson 2.7: THE UNDO COMMAND

```
   ** Press  u      to undo the last commands,   U  to fix a whole line. **

  1. Move the cursor to the line below marked ---> and place it on the
     first error.
  2. Type  x  to delete the first unwanted character.
  3. Now type  u  to undo the last command executed.
  4. This time fix all the errors on the line using the  x  command.
  5. Now type a capital  U  to return the line to its original state.
  6. Now type  u  a few times to undo the  U  and preceding commands.
  7. Now type CTRL-R (keeping CTRL key pressed while hitting R) a few times
     to redo the commands (undo the undo's).

---> Fiix the errors oon thhis line and reeplace them witth undo.

  8. These are very useful commands.  Now move on to the Lesson 2 Summary.




~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                           Lesson 2 SUMMARY


  1. To delete from the cursor up to the next word type:    dw
  2. To delete from the cursor to the end of a line type:    d$
  3. To delete a whole line type:    dd

  4. To repeat a motion prepend it with a number:   2w
  5. The format for a change command is:
               operator   [number]   motion
     where:
       operator - is what to do, such as  d  for delete
       [number] - is an optional count to repeat the motion
       motion   - moves over the text to operate on, such as  w (word),
               $ (to the end of line), etc.

  6. To move to the start of the line use a zero:  0

  7. To undo previous actions, type:           u  (lowercase u)
     To undo all the changes on a line, type:  U  (capital U)
     To undo the undo's, type:                 CTRL-R

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                     Lesson 3.1: THE PUT COMMAND


       ** Type      p  to put previously deleted text after the cursor. **

  1. Move the cursor to the first ---> line below.

  2. Type  dd  to delete the line and store it in a Vim register.

  3. Move the cursor to the c) line, ABOVE where the deleted line should go.

  4. Type   p   to put the line below the cursor.

  5. Repeat steps 2 through 4 to put all the lines in correct order.

---> d) Can you learn too?
---> b) Violets are blue,
---> c) Intelligence is learned,
---> a) Roses are red,
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                    Lesson 3.2: THE REPLACE COMMAND


       ** Type  rx  to replace the character at the cursor with  x . **

  1. Move the cursor to the first line below marked --->.

  2. Move the cursor so that it is on top of the first error.

  3. Type   r and then the character which should be there.

  4. Repeat steps 2 and 3 until the first line is equal to the second one.

--->  Whan this lime was tuoed in, someone presswd some wrojg keys!
--->  When this line was typed in, someone pressed some wrong keys!

  5. Now move on to Lesson 3.3.

NOTE: Remember that you should be learning by doing, not memorization.


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                    Lesson 3.3: THE CHANGE OPERATOR


          ** To change until the end of a word, type  ce . **

  1. Move the cursor to the first line below marked --->.

  2. Place the cursor on the  u  in  lubw.

  3. Type  ce  and the correct word (in this case, type  ine ).

  4. Press <ESC> and move to the next character that needs to be changed.

  5. Repeat steps 3 and 4 until the first sentence is the same as the second.

---> This lubw has a few wptfd that mrrf changing usf the change operator.
---> This line has a few words that need changing using the change operator.

Notice that  ce  deletes the word and places you in Insert mode.


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                    Lesson 3.4: MORE CHANGES USING c


     ** The change operator is used with the same motions as delete. **

  1. The change operator works in the same way as delete.  The format is:

         c    [number]   motion

  2. The motions are the same, such as   w (word) and  $ (end of line).

  3. Move to the first line below marked --->.

  4. Move the cursor to the first error.

  5. Type  c$  and type the rest of the line like the second and press <ESC>.
```

```
---> The end of this line needs some help to make it like the second.
---> The end of this line needs to be corrected using the  c$  command.

NOTE:  You can use the Backspace key to correct mistakes while typing.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                           Lesson 3 SUMMARY


  1. To put back text that has just been deleted, type   p .  This puts the
     deleted text AFTER the cursor (if a line was deleted it will go on the
     line below the cursor).

  2. To replace the character under the cursor, type   r   and then the
     character you want to have there.

  3. The change operator allows you to change from the cursor to where the
     motion takes you.  eg. Type  ce  to change from the cursor to the end of
     the word,  c$  to change to the end of a line.

  4. The format for change is:

        c   [number]   motion

Now go on to the next lesson.



~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
               Lesson 4.1: CURSOR LOCATION AND FILE STATUS

  ** Type CTRL-G to show your location in the file and the file status.
     Type  G  to move to a line in the file. **

  NOTE: Read this entire lesson before executing any of the steps!!

  1. Hold down the Ctrl key and press  g .  We call this CTRL-G.
     A message will appear at the bottom of the page with the filename and the
     position in the file.  Remember the line number for Step 3.

NOTE:  You may see the cursor position in the lower right corner of the screen
       This happens when the 'ruler' option is set (see  :help 'ruler'  )

  2. Press  G  to move you to the bottom of the file.
     Type  gg  to move you to the start of the file.

  3. Type the number of the line you were on and then  G .  This will
     return you to the line you were on when you first pressed CTRL-G.

  4. If you feel confident to do this, execute steps 1 through 3.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                    Lesson 4.2: THE SEARCH COMMAND


     ** Type  /  followed by a phrase to search for the phrase. **

  1. In Normal mode type the  /  character.  Notice that it and the cursor
     appear at the bottom of the screen as with the  :      command.

  2. Now type 'errroor' <ENTER>.  This is the word you want to search for.

  3. To search for the same phrase again, simply type  n .
     To search for the same phrase in the opposite direction, type  N .
```

```
  4. To search for a phrase in the backward direction, use  ?  instead of  / .

  5. To go back to where you came from press  CTRL-O  (Keep Ctrl down while
     pressing the letter o).  Repeat to go back further.  CTRL-I goes forward.

--->  "errroor" is not the way to spell error;  errroor is an error.
NOTE: When the search reaches the end of the file it will continue at the
      start, unless the 'wrapscan' option has been reset.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                Lesson 4.3: MATCHING PARENTHESES SEARCH


             ** Type  %  to find a matching ),], or } . **

  1. Place the cursor on any (, [, or { in the line below marked --->.

  2. Now type the  %  character.

  3. The cursor will move to the matching parenthesis or bracket.

  4. Type  %  to move the cursor to the other matching bracket.

  5. Move the cursor to another (,),[,],{ or } and see what  %  does.

---> This ( is a test line with ('s, ['s ] and {'s } in it. ))


NOTE: This is very useful in debugging a program with unmatched parentheses!



~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                   Lesson 4.4: THE SUBSTITUTE COMMAND


       ** Type  :s/old/new/g  to substitute 'new' for 'old'. **

  1. Move the cursor to the line below marked --->.

  2. Type  :s/thee/the <ENTER> .  Note that this command only changes the
     first occurrence of "thee" in the line.

  3. Now type  :s/thee/the/g .  Adding the  g  flag means to substitute
     globally in the line, change all occurrences of "thee" in the line.

---> thee best time to see thee flowers is in thee spring.

  4. To change every occurrence of a character string between two lines,
     type   :#,#s/old/new/g    where #,# are the line numbers of the range
                               of lines where the substitution is to be done.
     Type   :%s/old/new/g      to change every occurrence in the whole file.
     Type   :%s/old/new/gc     to find every occurrence in the whole file,
                           with a prompt whether to substitute or not.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                           Lesson 4 SUMMARY


  1. CTRL-G  displays your location in the file and the file status.
             G  moves to the end of the file.
     number  G  moves to that line number.
            gg  moves to the first line.

  2. Typing  /      followed by a phrase searches FORWARD for the phrase.
```

Typing ? followed by a phrase searches BACKWARD for the phrase.
After a search type n to find the next occurrence in the same direction
or N to search in the opposite direction.
CTRL-O takes you back to older positions, CTRL-I to newer positions.

3. Typing % while the cursor is on a (,),[,],{, or } goes to its match.

```
4. To substitute new for the first old in a line type    :s/old/new
   To substitute new for all 'old's on a line type      :s/old/new/g
   To substitute phrases between two line #'s type      :#,#s/old/new/g
   To substitute all occurrences in the file type       :%s/old/new/g
   To ask for confirmation each time add 'c'            :%s/old/new/gc
```

Lesson 5.1: HOW TO EXECUTE AN EXTERNAL COMMAND

** Type :! followed by an external command to execute that command. **

1. Type the familiar command : to set the cursor at the bottom of the screen. This allows you to enter a command-line command.

2. Now type the ! (exclamation point) character. This allows you to execute any external shell command.

3. As an example type ls following the ! and then hit <ENTER>. This will show you a listing of your directory, just as if you were at the shell prompt. Or use :!dir if ls doesn't work.

NOTE: It is possible to execute any external command this way, also with arguments.

NOTE: All : commands must be finished by hitting <ENTER>
From here on we will not always mention it.

Lesson 5.2: MORE ON WRITING FILES

** To save the changes made to the text, type :w FILENAME. **

1. Type :!dir or :!ls to get a listing of your directory.
You already know you must hit <ENTER> after this.

2. Choose a filename that does not exist yet, such as TEST.

3. Now type: :w TEST (where TEST is the filename you chose.)

4. This saves the whole file (the Vim Tutor) under the name TEST.
To verify this, type :!dir or :!ls again to see your directory.

NOTE: If you were to exit Vim and start it again with vim TEST , the file would be an exact copy of the tutor when you saved it.

5. Now remove the file by typing (MS-DOS): :!del TEST
or (Unix): :!rm TEST

Lesson 5.3: SELECTING TEXT TO WRITE

** To save part of the file, type v motion :w FILENAME **

1. Move the cursor to this line.

2. Press v and move the cursor to the fifth item below. Notice that the text is highlighted.

3. Press the : character. At the bottom of the screen :'<,'> will appear.

4. Type w TEST , where TEST is a filename that does not exist yet. Verify that you see :'<,'>w TEST before you press Enter.

5. Vim will write the selected lines to the file TEST. Use :!dir or !ls to see it. Do not remove it yet! We will use it in the next lesson.

NOTE: Pressing v starts Visual selection. You can move the cursor around to make the selection bigger or smaller. Then you can use an operator to do something with the text. For example, d deletes the text.

Lesson 5.4: RETRIEVING AND MERGING FILES

** To insert the contents of a file, type :r FILENAME **

1. Place the cursor just above this line.

NOTE: After executing Step 2 you will see text from Lesson 5.3. Then move DOWN to see this lesson again.

2. Now retrieve your TEST file using the command   :r TEST   where TEST is the name of the file you used.
   The file you retrieve is placed below the cursor line.

3. To verify that a file was retrieved, cursor back and notice that there are now two copies of Lesson 5.3, the original and the file version.

NOTE: You can also read the output of an external command. For example, :r !ls reads the output of the ls command and puts it below the cursor.

Lesson 5 SUMMARY

1. :!command executes an external command.

    Some useful examples are:

```
  (MS-DOS)      (Unix)
   :!dir               :!ls                -  shows a directory listing.
   :!del FILENAME   :!rm FILENAME   -  removes file FILENAME.
```

2. :w FILENAME writes the current Vim file to disk with name FILENAME.

3. v motion :w FILENAME saves the Visually selected lines in file FILENAME.

4. :r FILENAME retrieves disk file FILENAME and puts it below the cursor position.

5. :r !dir reads the output of the dir command and puts it below the cursor position.

Lesson 6.1: THE OPEN COMMAND

** Type o to open a line below the cursor and place you in Insert mode. **

1. Move the cursor to the line below marked --->.

2. Type the lowercase letter o to open up a line BELOW the cursor and place you in Insert mode.

3. Now type some text and press <ESC> to exit Insert mode.

---> After typing o the cursor is placed on the open line in Insert mode.

4. To open up a line ABOVE the cursor, simply type a capital O , rather than a lowercase o. Try this on the line below.

---> Open up a line above this by typing O while the cursor is on this line.

Lesson 6.2: THE APPEND COMMAND

** Type a to insert text AFTER the cursor. **

1. Move the cursor to the start of the line below marked --->.

2. Press e until the cursor is on the end of li .

3. Type an a (lowercase) to append text AFTER the cursor.

4. Complete the word like the line below it. Press <ESC> to exit Insert mode.

5. Use e to move to the next incomplete word and repeat steps 3 and 4.

---> This li will allow you to pract appendi text to a line.
---> This line will allow you to practice appending text to a line.

NOTE: a, i and A all go to the same Insert mode, the only difference is where the characters are inserted.

Lesson 6.3: ANOTHER WAY TO REPLACE

** Type a capital R to replace more than one character. **

1. Move the cursor to the first line below marked --->. Move the cursor to the beginning of the first xxx .

2. Now press R and type the number below it in the second line, so that it replaces the xxx .

3. Press <ESC> to leave Replace mode. Notice that the rest of the line remains unmodified.

4. Repeat the steps to replace the remaining xxx.

---> Adding 123 to xxx gives you xxx.
---> Adding 123 to 456 gives you 579.

NOTE: Replace mode is like Insert mode, but every typed character deletes an

```
       existing character.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                    Lesson 6.4: COPY AND PASTE TEXT


         ** Use the  y  operator to copy text and  p  to paste it **

  1. Go to the line marked with ---> below and place the cursor after "a)".

  2. Start Visual mode with  v  and move the cursor to just before "first".

  3. Type  y  to yank (copy) the highlighted text.

  4. Move the cursor to the end of the next line:  j$

  5. Type  p  to put (paste) the text.  Then type:  a second <ESC> .

  6. Use Visual mode to select " item.", yank it with  y , move to the end of
     the next line with  j$  and put the text there with  p .

--->  a) this is the first item.
      b)

  NOTE: you can also use  y  as an operator;  yw  yanks one word.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                        Lesson 6.5: SET OPTION


         ** Set an option so a search or substitute ignores case **

  1. Search for 'ignore' by entering:   /ignore  <ENTER>
     Repeat several times by pressing  n .

  2. Set the 'ic' (Ignore case) option by entering:   :set ic

  3. Now search for 'ignore' again by pressing  n
     Notice that Ignore and IGNORE are now also found.

  4. Set the 'hlsearch' and 'incsearch' options:  :set hls is

  5. Now type the search command again and see what happens:  /ignore <ENTER>

  6. To disable ignoring case enter:  :set noic

NOTE:  To remove the highlighting of matches enter:   :nohlsearch
NOTE:  If you want to ignore case for just one search command, use  \c
       in the phrase:  /ignore\c  <ENTER>
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                           Lesson 6 SUMMARY

  1. Type  o  to open a line BELOW the cursor and start Insert mode.
     Type  O  to open a line ABOVE the cursor.

  2. Type  a  to insert text AFTER the cursor.
     Type  A  to insert text after the end of the line.

  3. The  e  command moves to the end of a word.

  4. The  y  operator yanks (copies) text,  p  puts (pastes) it.

  5. Typing a capital  R  enters Replace mode until  <ESC>  is pressed.

  6. Typing ":set xxx" sets the option "xxx".  Some options are:
       'ic' 'ignorecase'   ignore upper/lower case when searching
```

```
       'is' 'incsearch'    show partial matches for a search phrase
       'hls' 'hlsearch'    highlight all matching phrases
     You can either use the long or the short option name.

  7. Prepend "no" to switch an option off:   :set noic

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                    Lesson 7.1: GETTING HELP


                   ** Use the on-line help system **

  Vim has a comprehensive on-line help system.  To get started, try one of
  these three:
       - press the <HELP> key (if you have one)
       - press the <F1> key (if you have one)
       - type   :help <ENTER>

  Read the text in the help window to find out how the help works.
  Type  CTRL-W CTRL-W   to jump from one window to another.
  Type    :q <ENTER>    to close the help window.

  You can find help on just about any subject, by giving an argument to the
  ":help" command.  Try these (don't forget pressing <ENTER>):

       :help w
       :help c_CTRL-D
       :help insert-index
       :help user-manual
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                   Lesson 7.2: CREATE A STARTUP SCRIPT


                      ** Enable Vim features **

  Vim has many more features than Vi, but most of them are disabled by
  default.  To start using more features you have to create a "vimrc" file.

  1. Start editing the "vimrc" file.  This depends on your system:
       :e ~/.vimrc         for Unix
       :e $VIM/_vimrc             for MS-Windows

  2. Now read the example "vimrc" file contents:
       :r $VIMRUNTIME/vimrc_example.vim

  3. Write the file with:
       :w

  The next time you start Vim it will use syntax highlighting.
  You can add all your preferred settings to this "vimrc" file.
  For more information type  :help vimrc-intro

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                         Lesson 7.3: COMPLETION


            ** Command line completion with CTRL-D and <TAB> **

  1. Make sure Vim is not in compatible mode:  :set nocp

  2. Look what files exist in the directory:  :!ls   or  :!dir

  3. Type the start of a command:  :e

  4. Press  CTRL-D  and Vim will show a list of commands that start with "e".
```

5. Press <TAB> and Vim will complete the command name to ":edit".

6. Now add a space and the start of an existing file name: :edit FIL

7. Press <TAB>. Vim will complete the name (if it is unique).

NOTE: Completion works for many commands. Just try pressing CTRL-D and <TAB>. It is especially useful for :help .

Lesson 7 SUMMARY

1. Type :help or press <F1> or <Help> to open a help window.

2. Type :help cmd to find help on cmd .

3. Type CTRL-W CTRL-W to jump to another window

4. Type :q to close the help window

5. Create a vimrc startup script to keep your preferred settings.

6. When typing a : command, press CTRL-D to see possible completions.
   Press <TAB> to use one completion.

This concludes the Vim Tutor. It was intended to give a brief overview of the Vim editor, just enough to allow you to use the editor fairly easily. It is far from complete as Vim has many many more commands. Read the user manual next: ":help user-manual".

For further reading and studying, this book is recommended:
     Vim - Vi Improved - by Steve Oualline
     Publisher: New Riders
The first book completely dedicated to Vim. Especially useful for beginners.
There are many examples and pictures.
See http://iccf-holland.org/click5.html

This book is older and more about Vi than Vim, but also recommended:
     Learning the Vi Editor - by Linda Lamb
     Publisher: O'Reilly & Associates Inc.
It is a good book to get to know almost anything you want to do with Vi.
The sixth edition also includes information on Vim.

This tutorial was written by Michael C. Pierce and Robert K. Ware, Colorado School of Mines using ideas supplied by Charles Smith, Colorado State University. E-mail: bware@mines.colorado.edu.

Modified for Vim by Bram Moolenaar.