

# Introduction to Vim, I

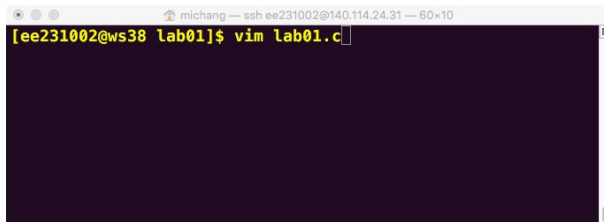
Introduction to Programming

EE231002

Sep. 26, 2016

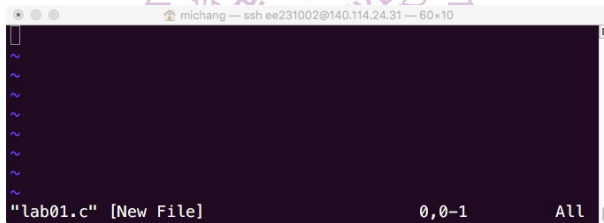
# Starting vim

- To start vim: `vim file`



A terminal window titled "michang — ssh ee231002@140.114.24.31 — 60x10". The prompt is "[ee231002@ws38 lab01]\$". The command "vim lab01.c" has been entered, and the cursor is at the end of the line.

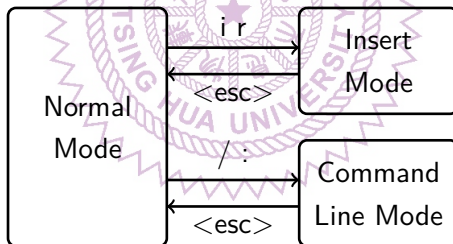
- For new file



A terminal window titled "michang — ssh ee231002@140.114.24.31 — 60x10". The prompt is "[ee231002@ws38 lab01]\$". The command "vim lab01.c" has been entered, and the cursor is at the end of the line. The terminal shows the vim editor in insert mode for a new file. The status bar at the bottom indicates "lab01.c" [New File], "0,0-1", and "All".

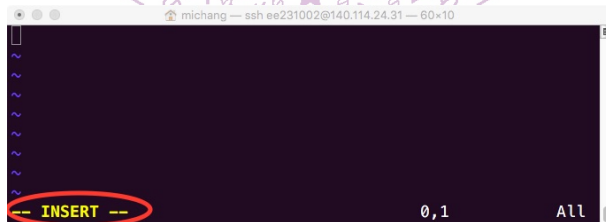
# Three Modes in vim

- There are three modes in **vim**
  - Normal mode: copy, delete, paste
  - Insert mode: insert text
  - Command line mode: save file, exit, search and replace















# Inserting Text

- When `vim` starts, it enters normal mode
- Press `i` to enter insert mode
  - Note the `-- INSERT --` on the lower-left corner
  - You can type in C program at this time



# Insert Mode

- In insert mode, you can type in texts
- To move cursor
  - , , ,  keys move cursor in four directions
  -  and  keys scroll one page of text
  -  key moves cursor to the beginning of the line
  -  key moves cursor to the end of the line
  -  key moves cursor to fixed columns (4x or 8x)
    - In our labs please use  key for indentation and each  key moves 4 spaces
- Press  key to return to normal mode

# Quitting vim

- In normal mode, the following commands save file or quit **vim** program
  - **:w**: save typed inputs to the file
  - **:q**: quit **vim** program (no saving file)
  - **:q!**: forced quitting from **vim** program
    - Changes are not updated to the file
  - **:wq**: save file and then quit **vim** program
  - **ZZ**: same as **:wq** but is a normal mode command
- Note that that the above except **ZZ** are executed in command line mode



The screenshot shows a terminal window with a dark background. At the top, the title bar reads "michang — ssh ee231002@140.114.24.112 — 60x10". The terminal displays a C program for temperature conversion. The code is as follows:

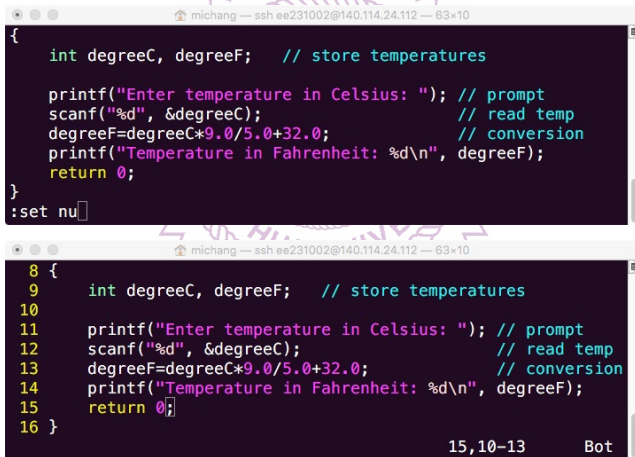
```
{
    int degreeC, degreeF;    // store temperatures

    printf("Enter temperature in Celsius: "); // prompt
    scanf("%d", &degreeC);                // read temp
    degreeF=degreeC*9.0/5.0+32.0;          // conversion
    printf("Temperature in Fahrenheit: %d\n", degreeF);
    return 0;
}
```

At the bottom of the terminal, the vim command line is visible, showing the command **:wq** entered in normal mode. The command is highlighted with a red circle.

# Show Line Numbers in vim

- vim does not show line numbers by default
  - Line numbers are very useful in debugging compiler errors
  - To show line number type in `:set nu` in normal mode



```
{
    int degreeC, degreeF;    // store temperatures

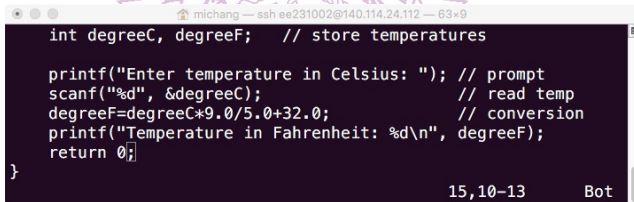
    printf("Enter temperature in Celsius: "); // prompt
    scanf("%d", &degreeC);                // read temp
    degreeF=degreeC*9.0/5.0+32.0;           // conversion
    printf("Temperature in Fahrenheit: %d\n", degreeF);
    return 0;
}
:set nu
```

```
8 {
9     int degreeC, degreeF;    // store temperatures
10
11     printf("Enter temperature in Celsius: "); // prompt
12     scanf("%d", &degreeC);                // read temp
13     degreeF=degreeC*9.0/5.0+32.0;           // conversion
14     printf("Temperature in Fahrenheit: %d\n", degreeF);
15     return 0;
16 }
15,10-13 Bot
```

# Color Text

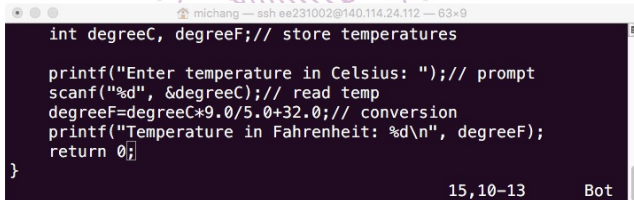
- **vim** takes advantage of the color terminal to make the file more legible
- The text color can be turned off by using `:syntax off` command
- `:syntax on` turns on color text



A screenshot of a terminal window titled "michang — ssh ee231002@140.114.24.112 — 63x9". The terminal displays C code with syntax highlighting: keywords are in blue, comments are in green, and strings are in red. The code is a function that takes Celsius temperature and returns Fahrenheit temperature. The status bar at the bottom right shows "15,10-13" and "Bot".

```
int degreeC, degreeF; // store temperatures

printf("Enter temperature in Celsius: "); // prompt
scanf("%d", &degreeC); // read temp
degreeF=degreeC*9.0/5.0+32.0; // conversion
printf("Temperature in Fahrenheit: %d\n", degreeF);
return 0;
}
```



A screenshot of a terminal window titled "michang — ssh ee231002@140.114.24.112 — 63x9". The terminal displays the same C code as the previous screenshot, but without syntax highlighting. The status bar at the bottom right shows "15,10-13" and "Bot".

```
int degreeC, degreeF;// store temperatures

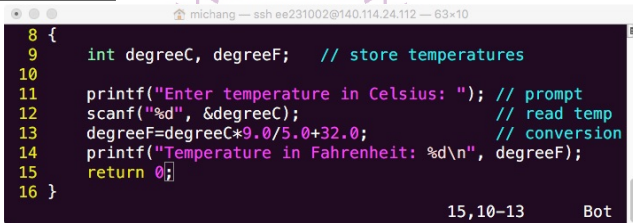
printf("Enter temperature in Celsius:");// prompt
scanf("%d", &degreeC);// read temp
degreeF=degreeC*9.0/5.0+32.0;// conversion
printf("Temperature in Fahrenheit: %d\n", degreeF);
return 0;
}
```



# Color Text, II

- Depending on terminal background, the text color may need to be adjusted


- `:set bg=dark`



A terminal window titled "michang — ssh ee231002@140.114.24.112 — 63x10" displays a C program. The text is color-coded: integers are yellow, keywords like 'int', 'printf', 'scanf', 'return', and 'break' are green, and comments are cyan. The program calculates Fahrenheit from Celsius. The status bar at the bottom right shows "15,10-13" and "Bot".

```
8 {  
9     int degreeC, degreeF;    // store temperatures  
10  
11     printf("Enter temperature in Celsius: "); // prompt  
12     scanf("%d", &degreeC);           // read temp  
13     degreeF=degreeC*9.0/5.0+32.0;    // conversion  
14     printf("Temperature in Fahrenheit: %d\n", degreeF);  
15     return 0;  
16 }
```

- `:set bg=light`

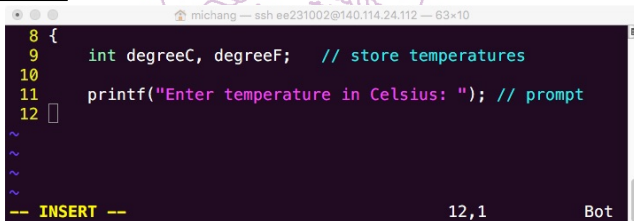


The same terminal window as above, but with the background color set to light. The syntax highlighting remains the same: integers are yellow, keywords are green, and comments are cyan. The status bar at the bottom right shows "15, 10-13" and "Bot".

```
8 {  
9     int degreeC, degreeF;    // store temperatures  
10  
11     printf("Enter temperature in Celsius: "); // prompt  
12     scanf("%d", &degreeC);           // read temp  
13     degreeF=degreeC*9.0/5.0+32.0;    // conversion  
14     printf("Temperature in Fahrenheit: %d\n", degreeF);  
15     return 0;  
16 }
```

# Auto-indent

- In insert mode, after typing a line of text the cursor moves to the first column – not aligned with the indented text
- This can be changed by `:set ai`, auto-indent, command
- `:set noai` sets no auto-indent



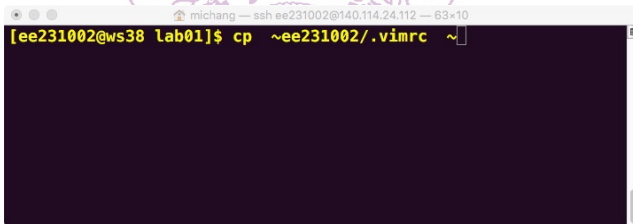
```
michang — ssh ee231002@140.114.24.112 — 63x10
8 {
9     int degreeC, degreeF;    // store temperatures
10
11     printf("Enter temperature in Celsius: "); // prompt
12
~
~
~
-- INSERT --                               12,1          Bot
```



```
michang — ssh ee231002@140.114.24.112 — 63x10
8 {
9     int degreeC, degreeF;    // store temperatures
10
11     printf("Enter temperature in Celsius: "); // prompt
12
~
~
~
-- INSERT --                               12,2-5        Bot
```

# .vimrc

- `vim` program executes the commands in `.vimrc` every time it is invoked.
- Please copy `~ee231002/.vimrc` to your home directory

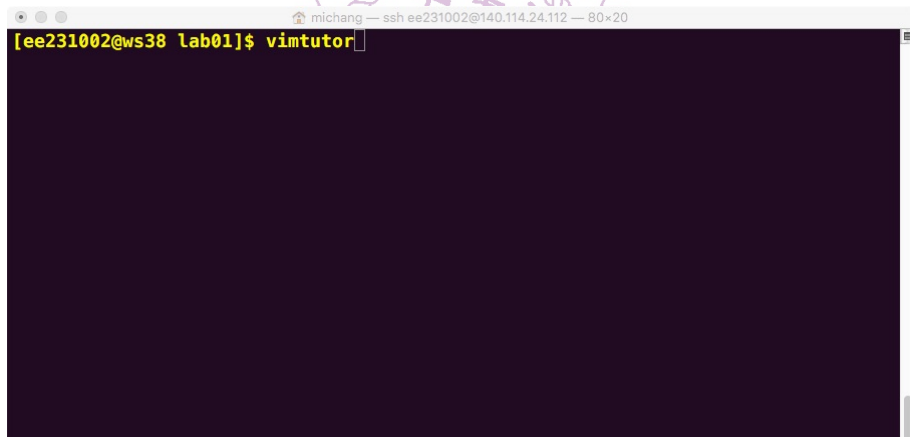


A terminal window with a title bar that reads "michang — ssh ee231002@140.114.24.112 — 63x10". The terminal prompt is "[ee231002@ws38 lab01]\$". The command being entered is "cp ~ee231002/.vimrc ~". The terminal background is dark purple, and the text is yellow.

- This file sets
  - Auto-indent mode
  - Each `Tab` inserts 4 spaces

# vim Tutorial

- `vim` program provides a tutorial for users to learn the easy commands
- At a linux terminal type in `vimtutor` as following to enter the tutorial

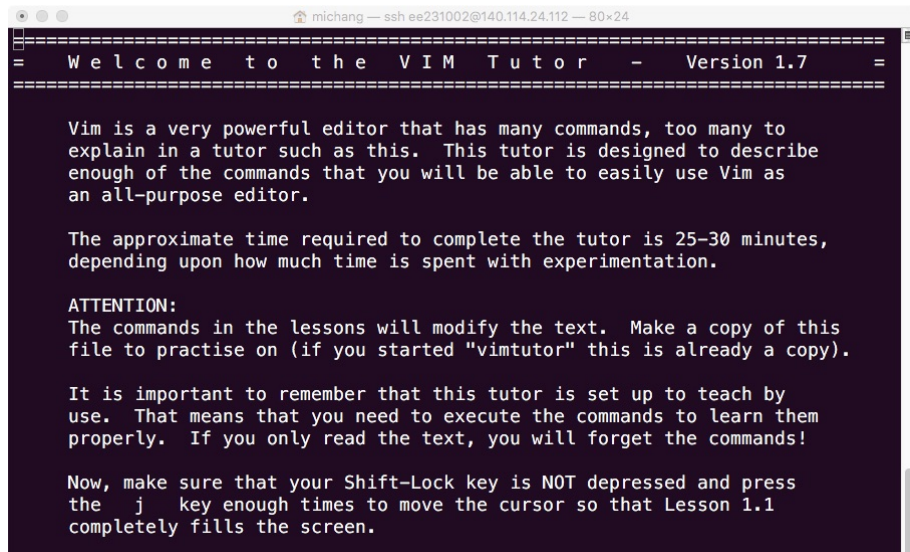


The screenshot shows a terminal window with a light gray title bar. The title bar text is "michang — ssh ee231002@140.114.24.112 — 80x20". The terminal content shows a prompt "[ee231002@ws38 lab01]" followed by the command "vimtutor" and a cursor. A large, faint, purple watermark of a sunburst with the Chinese characters "大華" (Dahua) is overlaid on the terminal area.

```
michang — ssh ee231002@140.114.24.112 — 80x20
[ee231002@ws38 lab01]$ vimtutor
```

# vim Tutorial, II

- Most frequently used commands are demonstrated



```
michang — ssh ee231002@140.114.24.112 — 80x24
=====
=  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.
```