

# vim命令圖解

The screenshot shows a Vim editor window with a Java code file. The code is as follows:

```

6 private boolean foo;
private boolean #ReallyLongName;
private boolean aReallyShortName;
private int bar1, bar2;

private boolean isFooAndBar(){
    foo = false;
}

{
public void main(String args[]){
    foo = true;
    if(foo){
0      bar1 = bar2 + 1 + fooClass.invokeRandomMethod();
      bar1 = bar2 + 2;
    }
    bar1++;
    bar2++;
    if(aReallyLongName
aReallyLongName
aReallyShortName
}
}
    
```

Annotations and callouts include:

- Line 6:** A green circle with the number 6.
- Character movement:** h, j, k, l, H, zt, Ctrl-B, M, zz, L, zb.
- Word movement:** w, W, b, B, e, W, B, e.
- Block movement:** fi, fd, %, {{, }}.
- Window management:** :split, :vsplit, :diffsplit, Ctrl-W p, Ctrl-W j, Ctrl-W k, Ctrl-W l.
- Search:** /fooCl, /fooCl.
- Other:** mx, 'x, Ctrl-N, gd, Ctrl-F.

## 游標移動/範圍單位

### 字元(character)

h j k l  
← ↓ ↑ →

### 單字(word)

w b 前/後個單字  
W B 前/後個單字(跳過符號)  
e 單字尾端

### 行(line)

O \$ 行頭 行尾  
^ 行頭(非空白字元)

### 段落(paragraph)、區塊(block)

{ } 上一段 下一段  
[[ ]] 區塊頭 區塊尾  
% 對應括號

### 螢幕(screen)、絕對位置

H 螢幕頂端 zt 捲至頂端  
M 螢幕中間 zz 捲至中間  
L 螢幕底部 zb 捲至底部  
C-B 上一頁 C-F 下一頁  
mx 標記x 'x 跳至標記x

### 搜尋(search)

\* # 向後/向前搜尋目前單字  
fx 向後搜尋字元x  
gd 跳至目前單字的定義位置  
/xxx 搜尋xxx  
n N 下/上一個搜尋結果

## 模式切換指令

ESC 進入normal mode  
v 進入visual mode  
V 進入visual line mode  
Ctrl-v 進入visual block mode  
i 進入insert mode  
R 進入replace mode  
a 在游標後附加  
A 在行末附加

## 動作指令

y 複製(範圍)  
d 刪除/剪下(範圍)  
c 修改(範圍)  
x 刪除/剪下(字元)  
D 刪除至行末  
C 修改至行末  
p 貼上  
J 和下一行合併  
r 替換(字元)  
> 縮排  
< 反縮排  
. 重複上一命令  
u 回復上一命令

## EX指令

:w 儲存(:wq 儲存並退出)  
:q 退出(:q! 強制退出)  
:e x 編輯檔案x  
:n 開新文件  
:h 呼叫vim help  
:xx 跳至xx行

## 自動補齊 [insert mode]

C-N C-P 自動補齊下/上個可能字  
C-X C-F 自動補齊可能檔名

## 分割視窗(split window)

:vsp :sp 垂直/水平分割視窗  
:diffs 分割視窗並比較(diff)檔案  
C-W p (來回)跳至前一個分割視窗  
C-W w 跳至下個分割視窗