

Key Term

Normal mode: The default mode in Vim where keys perform editor functions like movement rather than inserting text.

```
1 # Entering vim starts in normal mode
2 vim file.txt
3
4 # Pressing 'h' moves cursor left
5 # Pressing 'l' moves cursor right
```

Insert mode: Vim mode where keys insert text rather than performing commands. Entered with 'i', 'a', or 'o'.

```
1 # From normal mode
2 vim file.txt
3
4 # Press 'i' to enter insert mode
5 # Type text
6 # Press esc to exit insert mode
```

Visual mode: Vim mode that highlights text for cut/copy/paste. Entered with 'v', 'V', or Ctrl-V

```
1 # From normal mode
2 vim file.txt
3
4 # Press 'v' enter visual mode
5 # Move cursor to highlight text
6 # 'd' cuts text
7 # 'y' copies text
```

Ex commands: Commands run from Vim's command line rather than editor. Start with colon ':':

```
1 # From normal mode
2 :q # quit
3 :w # save
4 :%s/old/new # substitute all
```

Configuration: Customizing Vim behavior by setting options in ~/.vimrc file

```
1 # In ~/.vimrc
2 set nu # turn on line numbers
3 set ai # auto indent
4 syntax on # syntax highlighting
```

Introduction to VIM Normal Mode

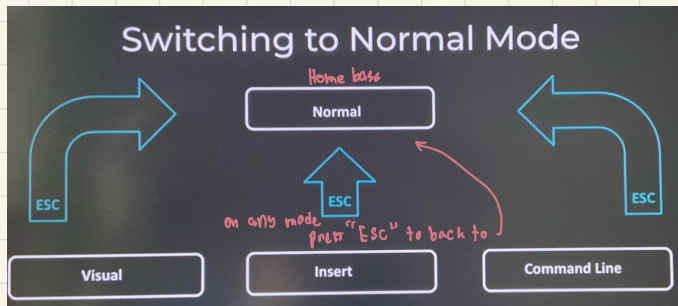
Vim uses only keyboard to do all of its commands, so you have to choose

VIM → on any Unix or Linux machine

- Text editor ^{only saves text}
- Based on **vi** ^{not word processor} ^{microsoft word → stores documents in a binary format}
- Keyboard only → no mouse, to click or move arc on keyboard ^{terminal}
- Customization through settings and plugins

Modes

- Normal ^{→ move around the file}
- Insert ^{→ type in text}
- Visual ^{→ executing, search, find, replace}
- Command line



Basic Navigation

to move the cursor

- h - move left
- j - move down
- k - move up
- l - move right

Extended Navigation

- w - word
- b - word beginning
- e - word end
- gg - top of file
- G - bottom of file

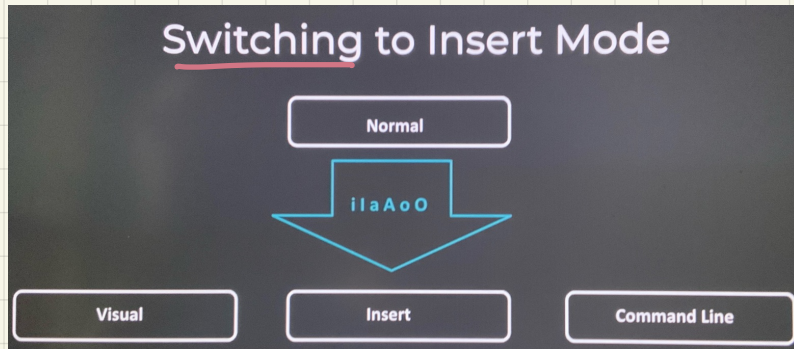
Yanking, Pasting, Deleting

called →

- y - copy highlighted text ^{→ to buffer}
- yw - copy word
- yy - copy line
- p - paste
- x - delete character
- dd - delete line

Switching from Normal to Insert and Visual Modes in VIM

Switching to Insert Mode

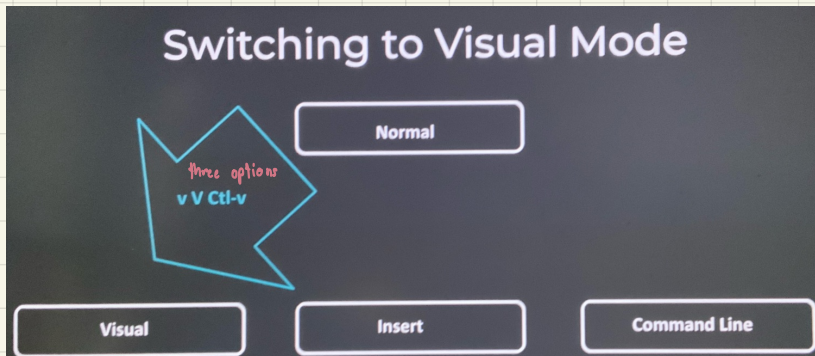


Entering Insert Mode

- i - before cursor
- I - before line
- a - after cursor
- A - end of line
- o - new line below
- O - new line above

specify the location to insert

Switching to Visual Mode



three options
v V Ctl-v

Entering Visual Mode

- v - by character
- V - by line
- Ctl-v - visual block

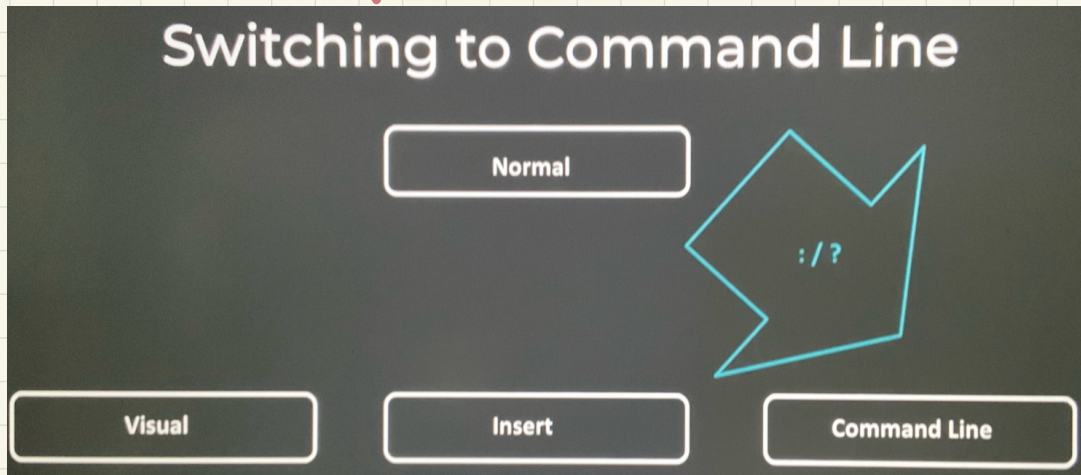
↳ after enter to this mode

↳ when you moved, it will highlight the text

↑
can use Yanking Here

Working with the VIM Command Line

Switching to Command Line



Entering Command Line

- : - execute command
- / - search down
- ? - search up
- :! - filter command

execute
↓
Command
:help
↳ type in command line

→ :e [file name]

Basic Commands

- help - documentation
- q - close buffer
- w - save buffer
- e - open file
- new - new buffer
- sav - save buffer as

Searching

- / - search ahead
- ? - search back
- n - next match
- N - previous match (go back)

```
28
29 import logging
30 import requests
31 import time
32 import json
33 from sets import Set
34 import urlparse
35
36 logging.basicConfig(format='%(asctime)s %(message)s',
37                     datefmt='%m/%d/%Y %I:%M:%S %p', level=logging.DEBUG)
38
39 #@TODO: maybe put this stuff in a config file
40 API_ENV = {"dev": "http://rest-dev.sqor.com/",
41            "stage": "https://rest-stage.sqor.com/",
42            "prod": "https://rest.sqor.com/",
43            "local": "http://localhost:8070/"}
44
45 API_PASSWORD = {"dev": {"email": "devops1@sqor.com",
46                        "password": "gfJoni0i2f3AvtM"},
47                 "prod": {"email": "devops1@sqor.com",
48                        "password": "pYjW6azHYhHR8oL"},
49                 "stage": {"email": "devops1@sqor.com",
50                        "password": "v8bPZTK2sPNAty"},
51                 "local": {"email": "noah@sqor.com", "password": "apiapi"}}
52
53 # find the word to replace
54 # if we want to do all instances
55 # :s/pass/cat/g → to do
```

Replace

find
pass

word to replace

if we want to do all instances
:s/pass/cat/g

VIM Configuration

```
python interface to Sqor REST API

http://rest.sqor.com/Help

Example usage:

In [1]: from sqor_api import SqorApi
In [2]: dev_api = SqorApi()
In [3]: nba = dev_api.fetch_athletes_by_sport("NBA")

In [5]: nba[0]
Out[5]:
{'birth_country': None,
 'first_name': 'u'Quincy',
 'team_id': 534,
 'team_name': 'u'Raptors',
 'weight': 225}
```

When starting VIM without configuration

Syntax Highlighting

```
Or for the Prod Environment:
```

```
In [7]: #nba_prod = prod_api.fetch_athl
```

```
In [8]: len(nba_prod)
```

```
Out[8]: 345
```

```
:syntax on
```

↙ Line Number

```
41 "stage": "https://r
42 "prod": "https://re
43 "local": "http://lo
44
45 API_PASSWORD = {"dev": {"email
46 "passw
:set number
```

New file

1

: new

Auto indent (when enter to new line)

```
1 def foo():
2     print(" ")
3
```

Python extension

```
def foo():  
    print("hello")
```

```
:sav cat.py
```

↓ saved to .py, so can be used

```
1 def foo():
2     print("hello")
3
4
```

text highlight from python

Location

[illegible]

show the location
on the line

Confic. commands

```
21
22 Plugin 'scroolose/nerdtre
23 Plugin 'corntrance/bufexplo
24 Plugin 'ervandew/supertab'
25 Plugin 'vim-scripts/taglis
"~/vimrc" 240 lines, 7538 bytes
```