

Objectives - 1/20/2022 (Thu)

- Test-driven development (TDD)
- VIM intermediate tips

Test-driven development

A method of development that helps you break the work into small increments that are tested as you go.

print_integer.c

~~source~~
implementation
function code

print_integer.h

header
function headers

expected.txt

expected output

test_print_integer.c

test code
int main() {

}

VIM - tabs

\$
↑
from
bash

vim -P

open
files
in
tabs

*,c *.h expected.txt

-OR-

vim -P *.*

:tabe
(^{OR}
tab edit)

expected.txt

AQ shortcut:
F2

VIM splits

↑
from
bash

vim

→ O
(capital
letter
O)

print_integer.*
~~print_it~~

→ OR →

Vertical

• VS
•
↑

print_integer, h

Use mouse
to adjust
split

↑
from
VIM

#

vim

→ O
(lowercase
letter
o)

print_integer.*

→ OR →

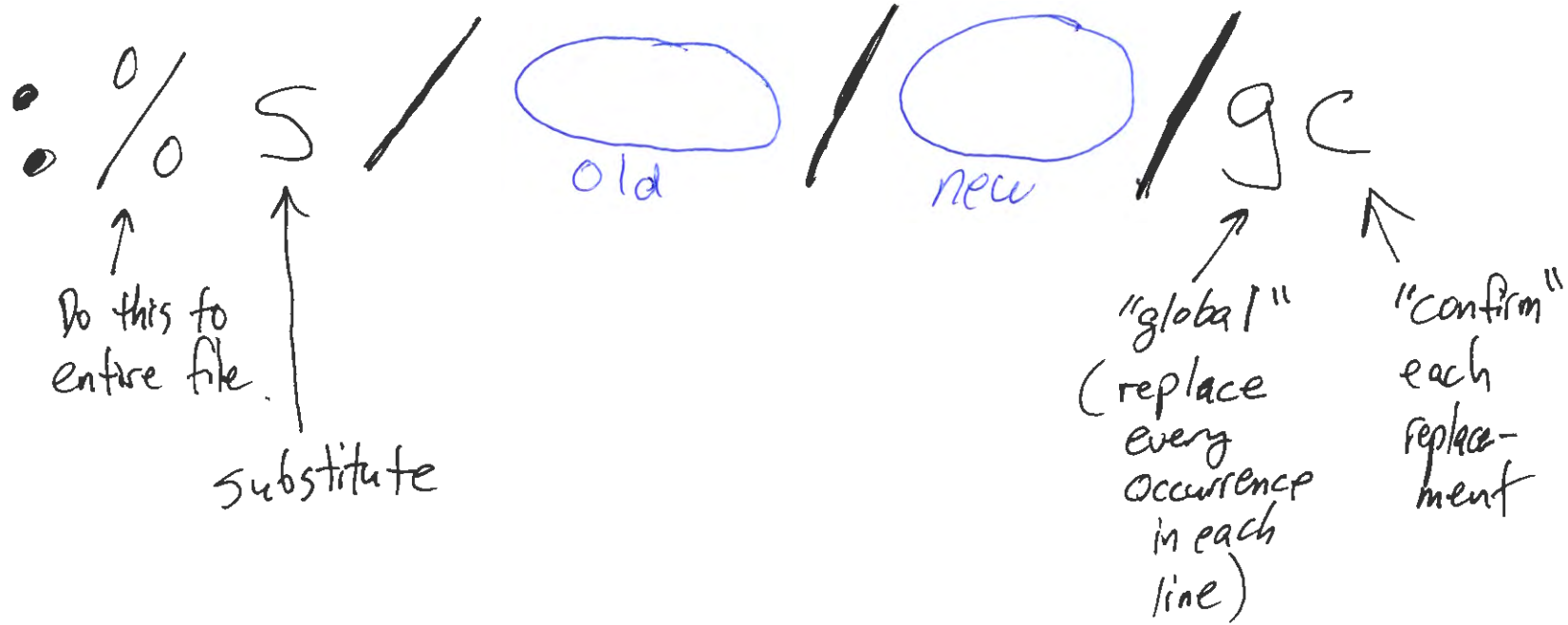
horizontal

• SP
•

print_integer, h

Vim - replace

Ugly way



Multiple cursors

`Ctrl-N` (for each occurrence)

`s` to substitute, i.e., start typing replacement

VIM - autocomplete

Ctrl-N

Ctrl-P

AQ shortcuts

F3

cycle through choices

F4

Example:

pr

Then, press Ctrl-N
to autocomplete to
print_integer

VIM Tips

Mouse is okay (except HwO1)

~~Chan~~

Replace:

:%s / (old) / (new) / g C

↑ ↑ ↑ ↑
entire substitute "global" "confirm"
document (all in each
 line) match

OR

Ctrl N (press for each occurrence)

S (substitute → start typing to replace)