

- The best way to learn Git (and other tools):
  1. Use it
  2. Find inability/inefficiency in your workflow
  3. Google a (better) solution
  4. Use it and build the muscle memory
- **Your gain = A constant x Your time spent on practicing**

# 1. Overview of Git

- What is Git?

A **distributed version control system**

Everyone has the whole commit history and acts like a server

- Three application situations (that I know):

- track the history of the development
- develop on multi-branches simultaneously
- Collaborative development

# 1. Overview of Git

What is Git?

A Version Control Tool.

Info in each commit:

SHA-1 checksum

Commit message

Time stamp

Creator

3. Merge two branches

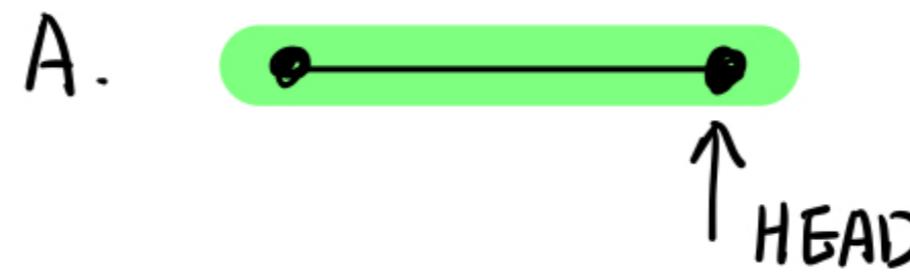
2. Create a branch

1. Commit new content

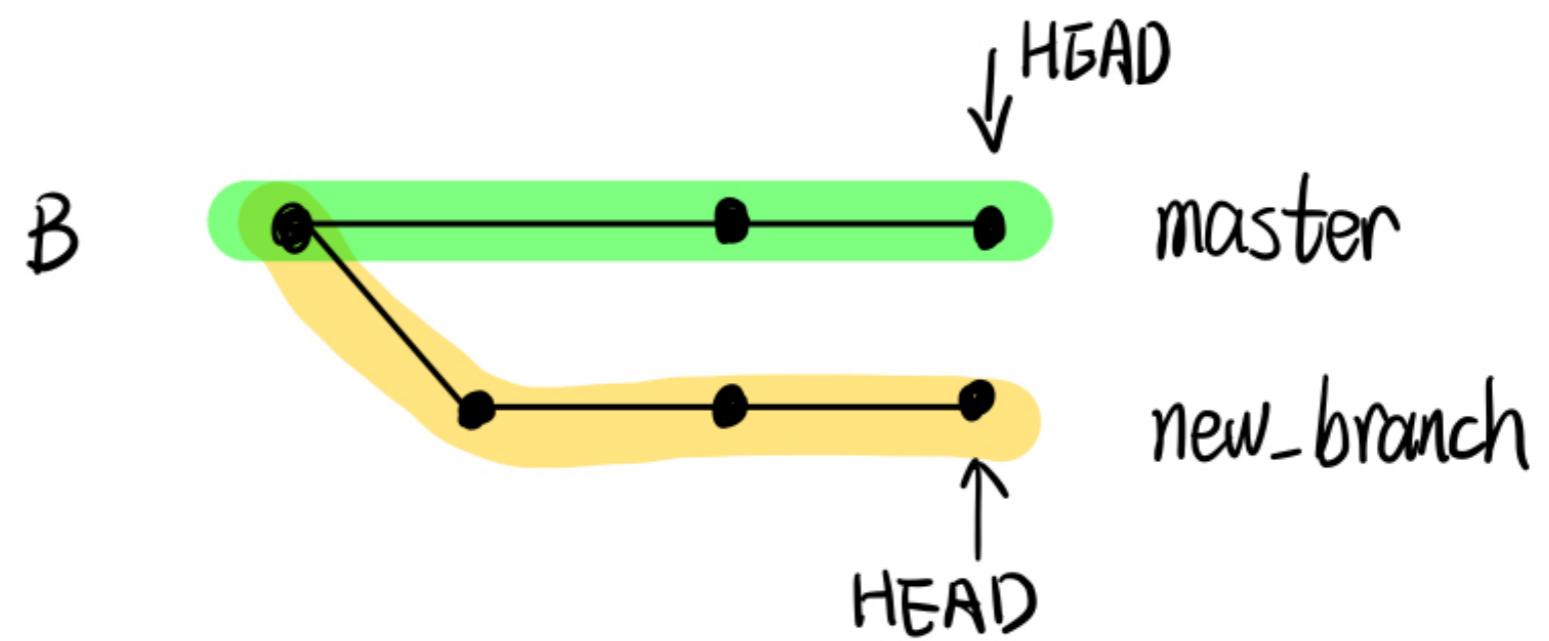
```
* 922d720 - (HEAD -> doc-tutorial, origin/doc-tutorial) modify the example of hyperslab (4 months ago) <Kosmos>
* 7a6d9dd - add using hyperslab examples (4 months ago) <Kosmos>
* ed8a6eb - Merge pull request #24 from ChenYangyao/doc-other (7 months ago) <Yangyao Chen>
| \
| * 045fc2f - change cmake install instructions (7 months ago) <Yangyao Chen>
| * 0d54a08 - avoid warning for init-order (7 months ago) <Yangyao Chen>
| * 01e8689 - avoid warning for sizeof (7 months ago) <Yangyao Chen>
| * 38efcf7 - reorganize cmake files (7 months ago) <Yangyao Chen>
| * 9fe4ff7 - Merge remote-tracking branch 'origin/dev' into doc-other (7 months ago) <Yangyao Chen>
| | \
| | /
| | /
| | *
| |   62d7c25 - Merge pull request #22 from ChenYangyao/doc-other (8 months ago) <Yangyao Chen>
| | |
| | *
| |   2fa3f96 - Merge pull request #20 from ChenYangyao/doc-other (8 months ago) <Yangyao Chen>
| | |
| |   * 8891d96 - remove ppt figures (7 months ago) <Yangyao Chen>
| |   * 6e4a51e - quick start - namespaces and namings (7 months ago) <Yangyao Chen>
| |   * 91b66ee - set numfig=True (7 months ago) <Yangyao Chen>
| |   * c3d036b - remove the gitbook figures (7 months ago) <Yangyao Chen>
| |   * 433b6ea - change example Makefile (7 months ago) <Yangyao Chen>
| |   * b1d1342 - add quick start examples (7 months ago) <Yangyao Chen>
| |   * a75ca5f - fix a typo (7 months ago) <Yangyao Chen>
| |   * 6571e7e - remove gitbook summary.md (7 months ago) <Yangyao Chen>
| |   * 5ec9552 - remove a badge for dev (7 months ago) <Yangyao Chen>
| | |
| |   * 957d595 - fix a typo in readme (8 months ago) <Yangyao Chen>
| | |
| | *
| |   f62e5fe - make doc tables of file modes and property list (8 months ago) <Yangyao Chen>
| | *
| |   56be33e - doc api-ref H5Dataset::create_attr (8 months ago) <Yangyao Chen>
| | *
| |   f2849fa - change test environment (8 months ago) <Yangyao Chen>
| | *
| |   b8e7bfd - release v0.1.0-alpha (8 months ago) <Yangyao Chen>
| | *
| |   33ef245 - add api-ref for H5File (8 months ago) <Yangyao Chen>
| | *
| |   bfd411f - doc the variants of create_attr (8 months ago) <Yangyao Chen>
| | *
| |   62e189c - add variants to create_attr (8 months ago) <Yangyao Chen>
| | *
| |   0c868ab - remove useless comments (8 months ago) <Yangyao Chen>
| | *
| |   b17f0e6 - reorder the interfaces for H5Dataset (8 months ago) <Yangyao Chen>
| | *
| |   a553162 - add H5Dataspace::create (8 months ago) <Yangyao Chen>
| | *
| |   43e15b1 - fix a typo for member access (8 months ago) <Yangyao Chen>
| | *
| |   335e38f - add class_t for H5Dataspace (8 months ago) <Yangyao Chen>
| | *
| |   7efea7d - remove useless comments (8 months ago) <Yangyao Chen>
| | *
| |   6bccea9 - change title to "I/O Arrays of Numeric Types" (8 months ago) <Yangyao Chen>
| | *
| |   bbdcf6e - add api-ref for h5 dataset, datatype, dataspace (8 months ago) <Yangyao Chen>
| | |
| |   * 2f3b8a6 - rtd version is set to 2 (8 months ago) <Yangyao Chen>
```

# 1. Overview of Git

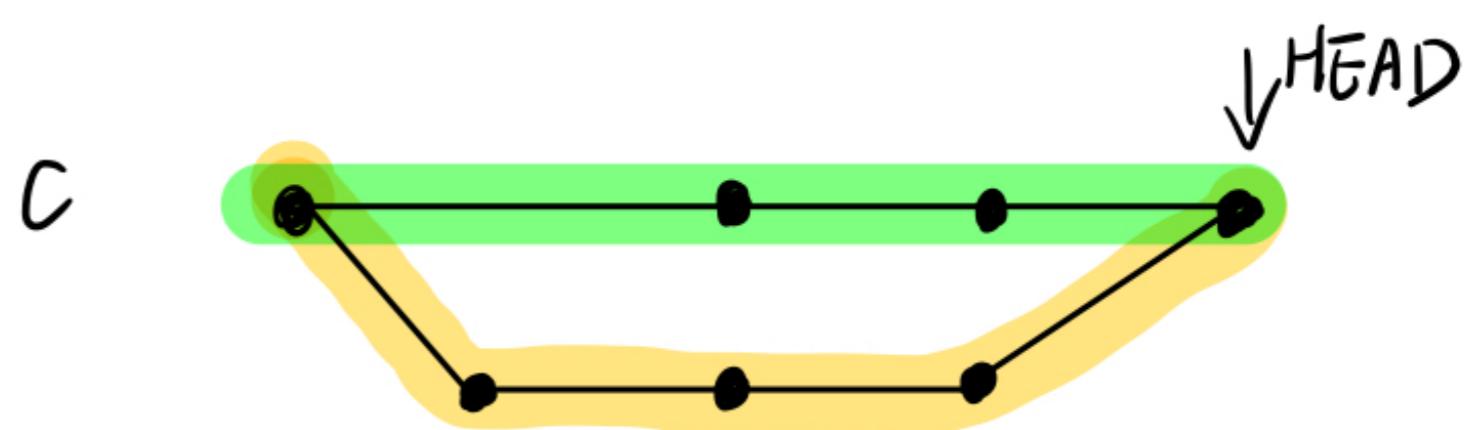
Commit



Branch

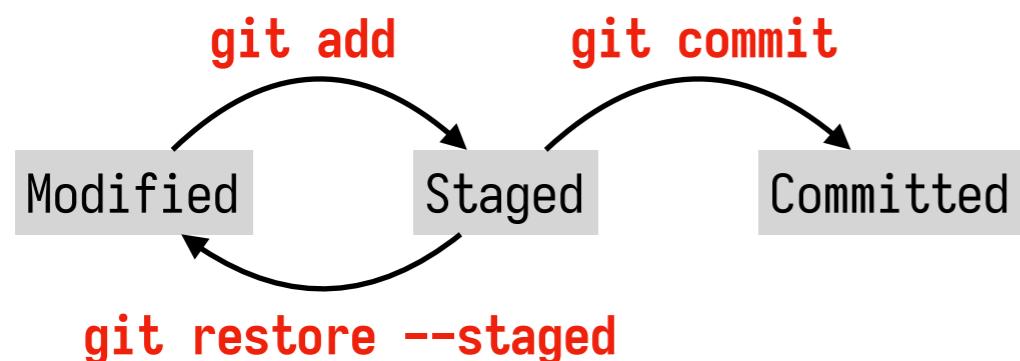


Merge



## 2. Change, stage and commit

- How to create a Git repository?
  1. Run `git init` in the terminal.
  2. Run `git clone some_remote_repo` to clone remote repository
- How to commit new content
  1. Modify files
  2. Stage the modified files (*Pack the modifications*)
  3. Commit the modification (*Commit message is necessary*)
- Three states of files for *tracked files*



```
~/Desktop/tester/git on master
> echo hello world >> README
~/Desktop/tester/git on master ?1
> cat README
hello world
~/Desktop/tester/git on master ?1
> git add README
~/Desktop/tester/git on master +1
> git commit -m "init the repo"
[master (root-commit) 5c94c30] init the repo
 1 file changed, 1 insertion(+)
 create mode 100644 README
~/Desktop/tester/git on master
> git status
On branch master
nothing to commit, working tree clean
```

- How to undo changes?
  - a. undo modified: `git restore filename`
  - b. undo staged (staged content will become *modified*):
    - I. `git reset HEAD -- filename`
    - II. `git restore --staged filename`
  - c. !!!undo committed: `git reset --hard some_check_sum` (All the commits after will be lost)
  - d. !!!clear modified and staged: `git reset --hard HEAD`

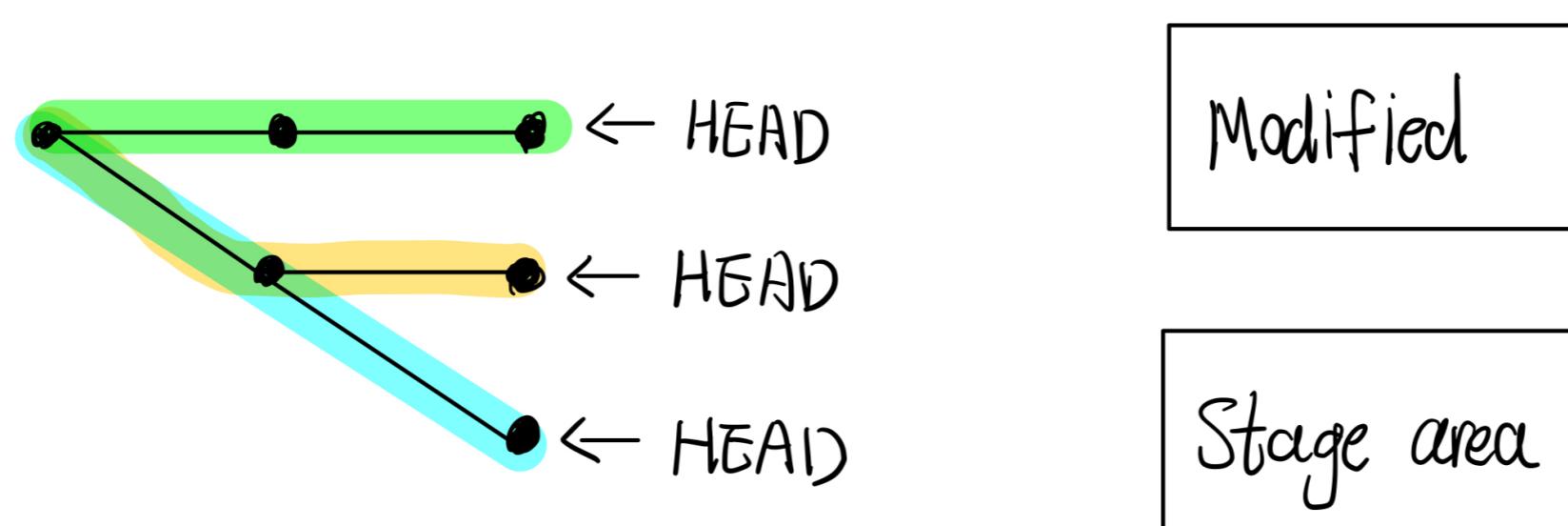
Why do we need stage operation?  
We may only want to commit part of the modification.

### 3. Branches

- How to create a new branch? `git branch branch_name`
- How to switch between branches? `git checkout branch_name`
- How to do them together? `git checkout -b branch_name`
- How to delete a branch? `git branch -d branch_name`
- How to merge other branch to the master one?
  1. `git checkout master`
  2. `git merge other_branch_name`

#### WARNING:

If there are some tracked-and-modified/staged content, you need to commit/stash them before you change the branch.



- Each branch has a HEAD pointer. And all the commits ahead of the HEAD pointer is undefined.
- All the branches share a common tracked-and-modified area and a common stage area.
- Stash will save the modified/staged content in a stack.

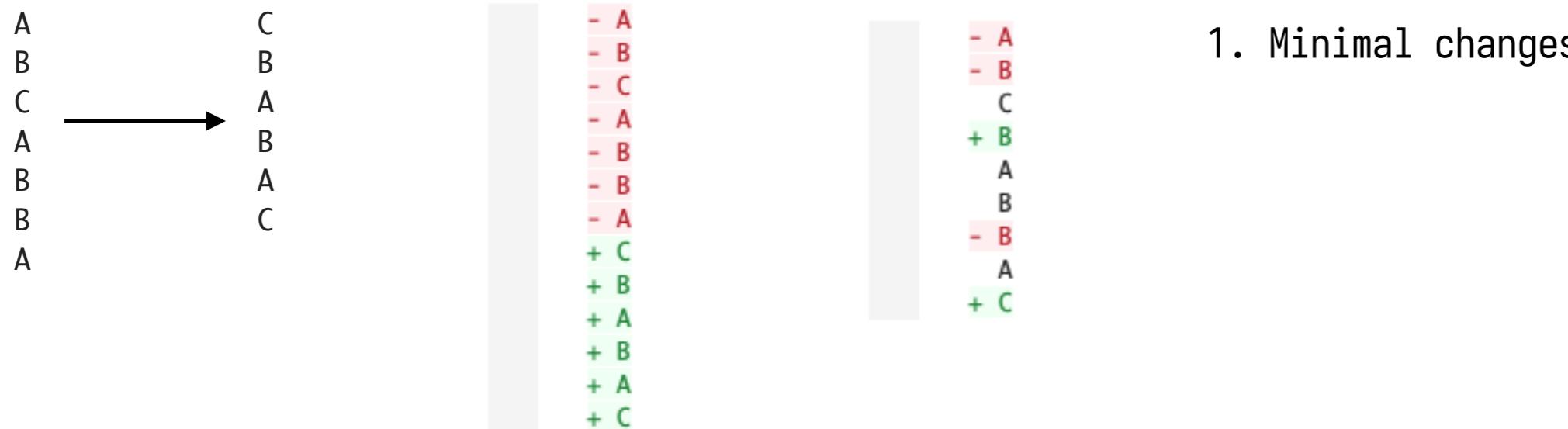
# 4. Git diff and merge

Application of `git diff`:

1. Commit new content
2. Merge two branches

--diff-algorithm={patience|minimal|histogram|myers}

Example:



Good: - one  
- two  
- three  
+ four  
+ five  
+ six

Bad: - one  
+ four  
- two  
+ five  
+ six  
- three

2. Delete first

Good: class Foo  
def initialize(name)  
@name = name  
end  
+  
+ def inspect  
+ @name  
+ end  
end

Bad: class Foo  
def initialize(name)  
@name = name  
+ end  
+  
+ def inspect  
+ @name  
end  
end

3. Follow code structure

## 4. Git diff and merge

```
17 @@ -198,10 +202,14 @@ void EigenField::calculate_eval_field(const TidalField3D &tf)
16     int total_size = side_num_ * side_num_ * side_num_;
15     eval_field_x_.resize(total_size);
14     auto &field_x = tf.field_x();
13 -    /* #pragma omp parallel for num_threads(threads_) */
12 -    for (int i = 0; i < total_size; ++i) {
11 -        es_.compute(symm_mat2eigen_mat(field_x[i]), false);
10 -        assign_eigen_val(es_, eval_field_x_[i]);
9 +#pragma omp parallel num_threads(threads_)
8 +
7 +    Eigen::EigenSolver<Eigen::Matrix3d> es;
6 +#pragma omp for private(es)
5 +
4 +    for (int i = 0; i < total_size; ++i) {
3 +
2 +        es.compute(symm_mat2eigen_mat(field_x[i]), false);
1 +
}
```

Hunk: one set of changes

-198, 10: line [198, 198 + 10) in the old file

+202, 14: line [202, 202 + 14) in the new file

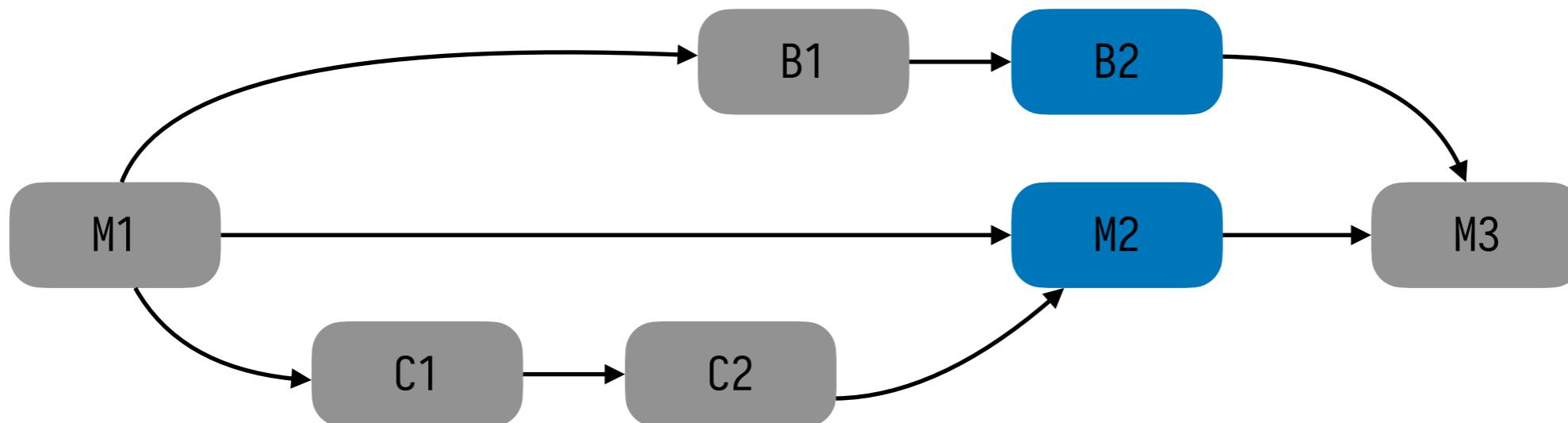
-red: deleted content

+green: added content

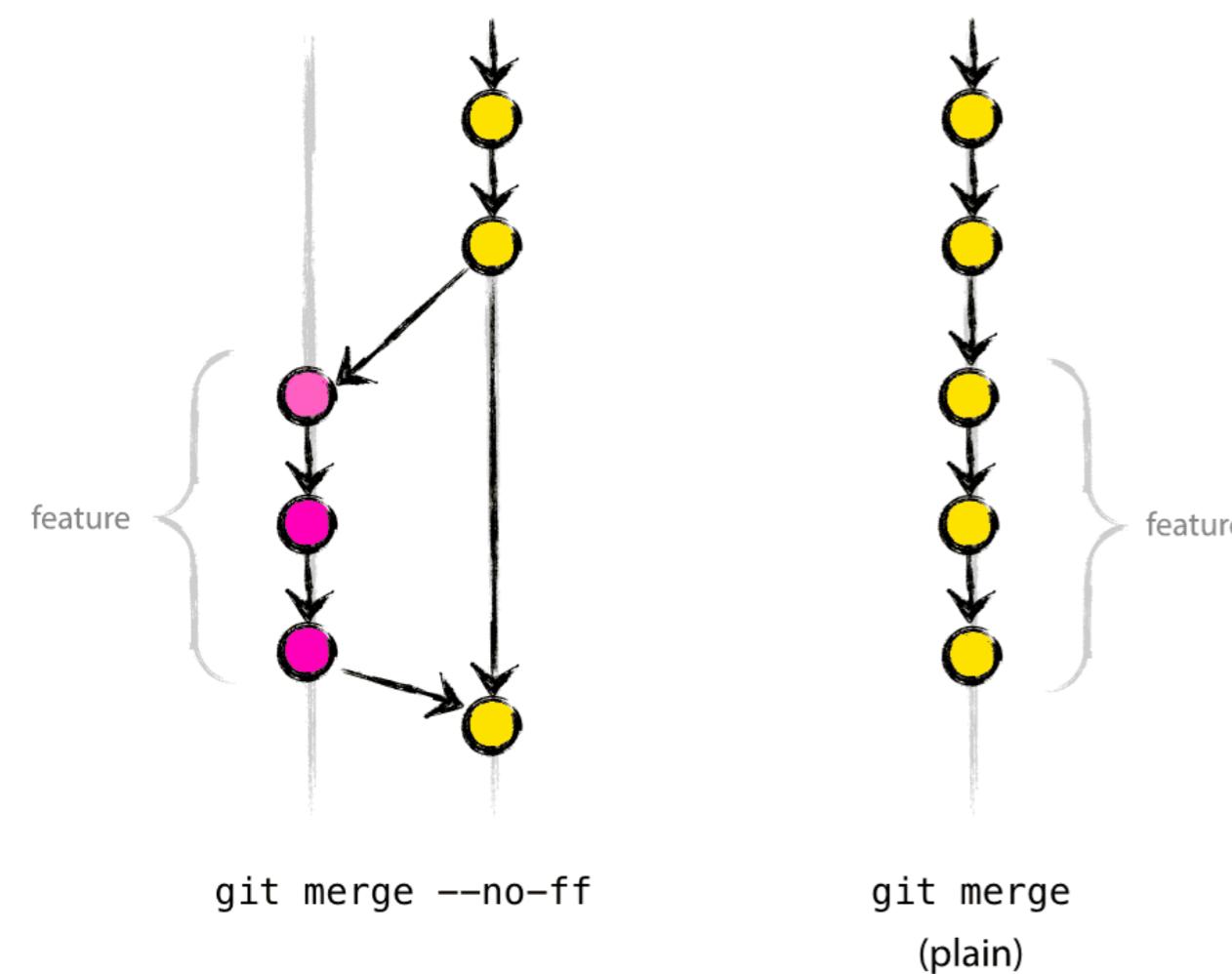
Git diff:

1. between modified and HEAD: `git diff <filename>`
2. between modified and any commit: `git diff <commit> <filename>`
3. between staged and any commit: `git diff --cached <commit> <filename>`
4. between two commits: `git diff <commit_1> <commit_2> <filename>`

## 4. Git diff and merge



Merge C2 with M1 to get M2

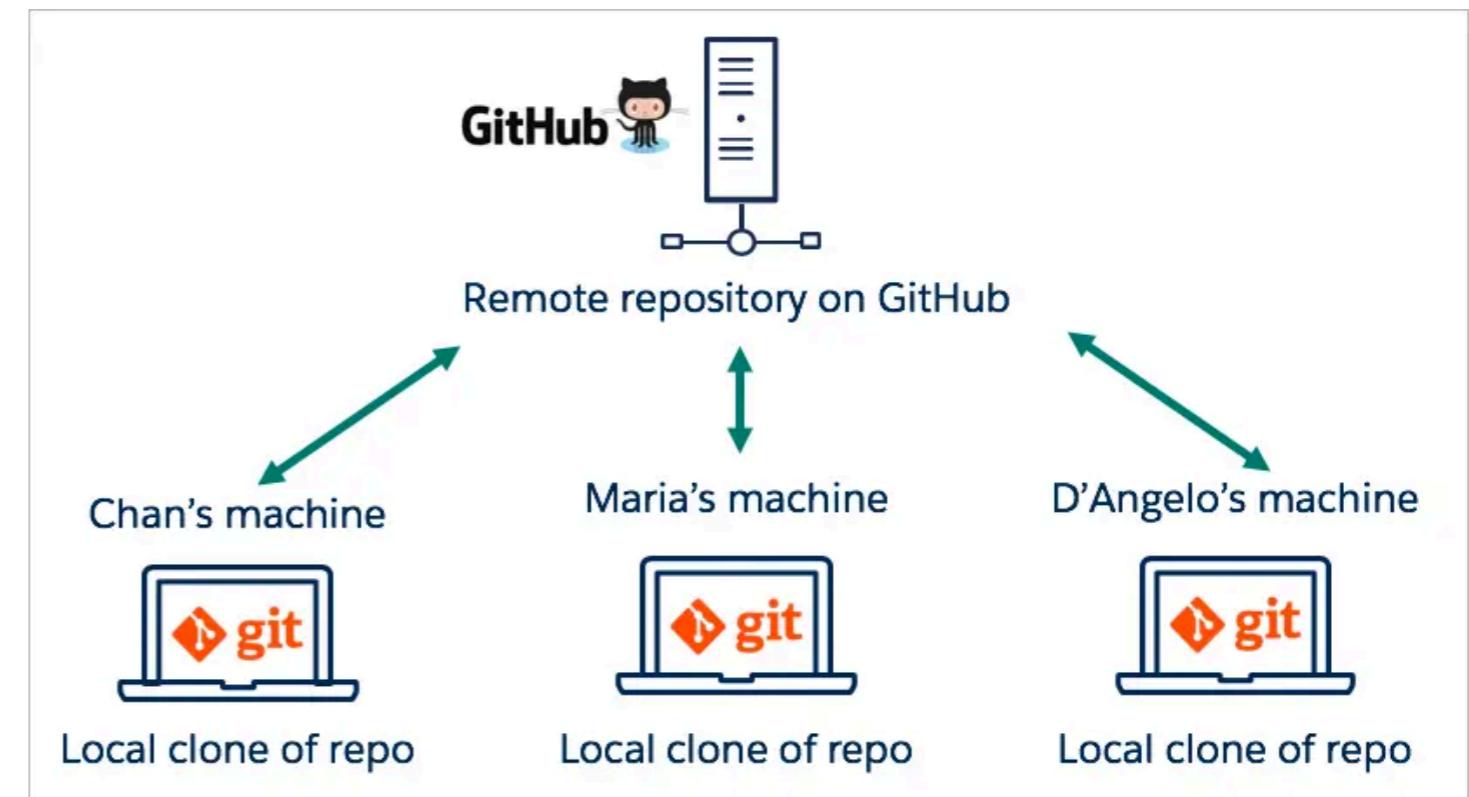


Merge B2 with M2 to get M3?  
(Three-way merge)

1. find their most close common ancestor: M1
2. **git diff M1 B2**
3. **git diff M1 M3**
4. compare each hunk
  - same changes  $\Rightarrow$  accept
  - different changes  $\Rightarrow$  conflict

# 5. Remote repository

- How to link to a remote repo?
  1. `git clone remote repo`
  2. `git remote add origin git@github.com:ChenYangyao/hipp.git`
- How to sync with the remote repo?
  1. `git fetch origin`
  2. `git merge origin` (fetch + merge = pull)
  3. fix conflicts if there are any
  4. `git push (remote_repo) (branch)`
- Tips:
  - Do not do changes directly on remote repo;
  - Interact with remote repo only though push/pull;
  - Always do pull before push;
  - Keep a clean master branch;



## 6. gitignore



```
1 my_setting
1 *.o
2 *.out
3 */*.o
4 */*.out
5 *.pdf
6 *.hdf5
7 *.h5
8 */*.out
9 */*.cccls/*
10 .vscode
11 .pytest_cache/
12 .vim/
13 *.vim
14 tmp/
15 __pycache__/
16 */__pycache__/
17 *.pyc
18 .DS_Store
19 .ipynb_checkpoints/
20 tags
21 *.log
22 .gitignore
23 .sync_repo.conf
24 *.ipynb
25 # ignore all the result file
26 */result/
27 data/
28 note/
29 paper/
30 reference/
31 report/
32 result/
33 _halo_based_group_finder_cpp/
34
35
36 # Byte-compiled / optimized / DLL files
37 *.py[cod]
38 *$py.class
39
40 # C extensions
41 *.so
42
43 # Distribution / packaging
A ① .gitignore 1:1 (0% 156L)
```

Ignore unnecessary files,  
especially large ones!  
e.g. data file, executable files etc.

All committed files will be in the  
repository permanently.

# 6. Plugins in Neovim

tpope/vim-fugitive

```
5 Head: master
4 Merge: origin/master
3 Help: g?
2
1 Unstaged (3)
6 M nvim/lua/gitsigns_nvim.lua
1 @@ -11,8 +11,8 @@ require('gitsigns').setup {
2   keymaps = {
3     -- Default keymap options
4     noremap = true,
5     ['n ]c'] = {expr = true, "&diff ? ']h' : '<cmd>lua require\"gitsigns\".next_hunk()'},
6     ['n [c'] = {expr = true, "&diff ? '[h' : '<cmd>lua require\"gitsigns\".prev_hunk()'"},
7     ['n ]h'] = {expr = true, "&diff ? ']h' : '<cmd>lua require\"gitsigns\".next_hunk()'},
8     ['n [h'] = {expr = true, "&diff ? '[h' : '<cmd>lua require\"gitsigns\".prev_hunk()'},
9     ['n <leader>hs'] = '<cmd>lua require"gitsigns".stage_hunk()',
10    ['v <leader>hs'] = '<cmd>lua require"gitsigns".stage_hunk({vim.fn.line("."), vim.fn.line("v")})',
11    ['n <leader>hu'] = '<cmd>lua require"gitsigns".undo_stage_hunk()',
12 M nvim/lua/nvim_cmp.lua
13 M nvim/vimplug.vim
14
15 Staged (1)
16 M nvim/keymapping.vim
```

A ≡ .git/index [-] 1:6 (27% 22L)

# 6. Plugins in Neovim

## tpope/vim-fugitive

The image shows a Vim session with two panes, A and B, side-by-side. Both panes display the same code content, which is a Vim configuration file for managing plugins.

```
+ 10 ---109 lines: vim plugins manager:vim-plug--- + 9 ---109 lines: vim plugins manager:vim-plug---  
9 Plug 'derekwyatt/vim-fswitch', {'for': ['c', 'cpp', 'h', 'hpp']}  
8  
7 "  
6 " Git  
5 "  
4 " show git modified line in the state column  
3 " Plug 'airblade/vim-gitgutter'  
2 Plug 'lewis6991/gitsigns.nvim'  
1 " powerful git tool  
119 Plug 'tpope/vim-fugitive'  
1 " tree like git browser  
2 Plug 'junegunn/gv.vim', { 'on': 'GV' }  
3 " github  
4 --- 43 lines: Plug 'pwntester/octo.nvim'..... + 5 Plug 'hrsh7th/cmp-buffer'  
5 Plug 'hrsh7th/nvim-cmp'  
6 Plug 'kdheepak/cmp-latex-symbols', {'for': ['tex']}  
7 Plug 'quangnguyen30192/cmp-nvim-ultisnips'  
8 Plug 'hrsh7th/cmp-path'  
9 Plug 'hrsh7th/cmp-nvim-lua'  
10 Plug 'lukas-reineke/cmp-under-comparator'  
11 "  
12 "  
13 " Lua  
14 "  
15 Plug 'RishabhRD/popfix', {'for': ['lua']}  
16 Plug 'wkcsmology/vim-lua-format', {'for': ['lua']}  
17 --- 28 lines:  
18 --- 28 lines:  
19 "
```

The code lists several plugin configurations. Notable entries include:

- Line 119: `Plug 'tpope/vim-fugitive'`
- Line 120: `1 " tree like git browser`
- Line 121: `2 Plug 'junegunn/gv.vim', { 'on': 'GV' }`
- Line 122: `3 " github`
- Line 123: `4 --- 43 lines: Plug 'pwntester/octo.nvim'.....`
- Line 124: `5 Plug 'hrsh7th/cmp-buffer'
- Line 125: `6 Plug 'hrsh7th/nvim-cmp'
- Line 126: `7 Plug 'kdheepak/cmp-latex-symbols', {'for': ['tex']}
- Line 127: `8 Plug 'quangnguyen30192/cmp-nvim-ultisnips'
- Line 128: `9 Plug 'hrsh7th/cmp-path'
- Line 129: `10 Plug 'hrsh7th/cmp-nvim-lua'
- Line 130: `11 Plug 'lukas-reineke/cmp-under-comparator'
- Line 131: `12 "  
13 "  
14 " Lua  
15 "  
16 Plug 'RishabhRD/popfix', {'for': ['lua']}  
17 Plug 'wkcsmology/vim-lua-format', {'for': ['lua']}  
18 --- 28 lines:  
19 "

The Vim interface includes status bars at the bottom for both panes, showing file names, line numbers, and other status information. The background of the Vim window features a dark illustration of a character from the video game Okami.

# 6. Plugins in Neovim

## tpope/vim-fugitive

The screenshot shows a Neovim window with three panes (A, B, C) displaying commit history from a Git repository. The commits are color-coded by author:

- Yangyao (blue):**
  - commit 4: hello world
  - commit 3: this is the committed message
  - commit 2: I deleted Yangyao and Kai's code
  - commit 1: And I add three of mine
  - commit 5: hahaha
- Kai (orange):**
  - commit 4: hello world
  - commit 3: this is the committed message
  - commit 2: <<<< HEAD
  - commit 1: I deleted Yangyao and Kai's code
  - commit 5: And I add three of mine
  - commit 1: hahaha
  - commit 2: =====
  - commit 3: Yangyao want to add this line
  - commit 4: This is branch-dev
  - commit 5: >>>> dev
- Wangk (green):**
  - commit 3: hello world
  - commit 2: this is the committed message
  - commit 1: Yangyao want to add this line
  - commit 4: This is branch-dev

The fugitive.vim plugin highlights specific lines of code in each commit message. In pane A, lines 2 and 5 are highlighted. In pane B, lines 2, 4, and 5 are highlighted. In pane C, lines 1 and 4 are highlighted.

At the bottom of the screen, the status bar shows:

- Panels: A, B, C
- File: README
- Mode: README
- Line: 7:5<L
- Cursor: 0 0
- Registers: A 0
- Buffer: master +7
- File: README
- Panel: C

Bottom right corner: 12:25:48 2021-10-23

# 6. Plugins in Neovim

## tptope/vim-fugitive

```
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 36 " devicons
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 35 Plug 'ryanoasis/vim-devicons'
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 34 " choose the window
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 33 Plug 't9md/vim-choosewin',
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 32 " start up window
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 31 Plug 'mhinz/vim-startify'
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 30
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 29 "
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 28 " C/C++ related
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 27 "
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 26 " switch header and source file
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 25 Plug 'derekwyatt/vim-fswitch', {'for': ['c', 'cpp', 'h', 'hpp']}
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 24
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 23 "
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 22 " Git
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 21 "
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 20 " show git modified line in the state column
d3808cd9 (Kosmos 2021-10-19 09:50:23 -0400) 19 Plug 'lewis6991/gitsigns.nvim'
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 18 " powerful git tool
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 17 Plug 'tpope/vim-fugitive'
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 16 " tree like git browser
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 15 Plug 'junegunn/gv.vim', { 'on': 'GV' }
9379eb23 (Kosmos 2021-04-11 10:19:11 -0400) 14 " github
9379eb23 (Kosmos 2021-04-11 10:19:11 -0400) 13 Plug 'pwntester/octo.nvim'
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 12
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 11 "
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 10 " Anxuliary
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 9 "
015bd999 (Kosmos 2021-03-10 19:41:57 -0500) 8 " useful plugin for lua
015bd999 (Kosmos 2021-03-10 19:41:57 -0500) 7 Plug 'nvim-lua/plenary.nvim'
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 6 " Great terminal tool
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 5 Plug 'voldikss/vim-floaterm', { 'on': ['FloatermNew', 'Floaterms', 'FloatermToggle']}
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 4 " asynchronously check error
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 3 Plug 'dense-analysis/ale'
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 2 " add head information
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 1 Plug 'alpertuna/vim-header', { 'on': 'AddHeader'}
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 135 " run the test file, support many filetypes, including python
9379eb23 (Kosmos 2021-04-11 10:19:11 -0400) 1 " Plug 'vim-test/vim-test', {'on': ['TestNearest', 'TestFile']}
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 2 " task control
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 3 Plug 'skywind3000/asynctasks.vim'
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 4 Plug 'skywind3000/asyncrun.vim'
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 5 " vimtex
11b0a47f (Kosmos 2021-03-08 12:05:34 -0500) 6 Plug 'lervag/vimtex'
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 7 " spell check
9379eb23 (Kosmos 2021-04-11 10:19:11 -0400) 8 Plug 'kamykn/spelunker.vim', { 'for': ['tex', 'markdown', 'html', 'json'] }
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 9 " gist
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 10 Plug 'lambdalisue/vim-gista', { 'on': ['Gista'] }
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 11 " Cheatsheet for vim
^1ba5e5e (Kosmos 2021-03-02 22:51:57 -0500) 12 Plug 'lifepillar/vim-cheat40', { 'on': ['Cheat'] }
```

A 8,fugitiveblame [-] 1:135<5% 205L) 0 0 ▲ 0

:Git blame

wangk@Kosmos ~

B /Users/wangk/.config/nvim/vimplug.vim

0:nvim\*

① 12:28:08 2021-10-23

# 6. Plugins in Neovim

## lewis6991/gitsigns.nvim

```
1 " -----
2 " C/C++ related
3 " -----
4 " switch header and source file
5 Plug 'derekwyatt/vim-fswitch', {'for': ['c', 'cpp', 'h', 'hpp']}
6 sdfk
7 sdfjkl
8 sdf1kjd
9
10 " -----
11 " Git
12 " -----
13 " show git modified line in the state column
14 Plug 'lewis6991/gitsigns.nvim'
15 " powerful git tool
16 Plug 'tpope/vim-fugitive'
17 " tree like git browser
18 Plug 'junegunn/gv.vim', { 'on': 'GV' }
```

# 6. Plugins in Neovim

junegunn/gv.vim

```
1. [Default] 2. [Default]
1  2021-10-22 27487b1 (HEAD -> doc-tutorial) Merge branch 'dev' into doc-tutorial (Kosmos)
1 \
2 | * 2021-10-02 5afbfcb (origin/dev, dev) Merge pull request #64 from ChenYangyao/doc-other (Yangyao Chen)
3 |
4 | | * 2021-10-02 42897d4 (origin/doc-other) Use ConstDatapacket for pack operation (Yangyao Chen)
5 | | * 2021-10-02 f0808f0 add mpi examples in api-ref (Yangyao Chen)
6 | | * 2021-10-02 1acae6e new implementation for move ops (Yangyao Chen)
7 | | * 2021-10-02 f187729 use ConstDatapacket for const buffer (Yangyao Chen)
8 | | * 2021-10-02 8c55fa9 use ConstDatapacket for send/push (Yangyao Chen)
9 | | * 2021-10-02 a62b369 use new Datapacket interface (Yangyao Chen)
10 | | * 2021-10-02 654de96 move the comment string (Yangyao Chen)
11 | | * 2021-10-02 d1db8d6 new Datapacket implementation (Yangyao Chen)
12 | | * 2021-10-02 fc7ca6a new TypeCvt interface (Yangyao Chen)
13 | | * 2021-10-02 730bf42 new TypeCvt implementation (Yangyao Chen)
14 | | * 2021-09-30 22c33df new datapacket interface and implementation (Yangyao Chen)
15 | | * 2021-09-30 c6c1a50 make table tighter (Yangyao Chen)
16 | | * 2021-09-30 8a84982 add a const (Yangyao Chen)
17 | | * 2021-09-30 5475b9f adopt new datapacket design - separate the const (Yangyao Chen)
18 | | * 2021-09-29 4b67089 implement the new datatype and datapacket API (Yangyao Chen)
19 | | * 2021-09-29 f1a92ad init env test (Yangyao Chen)
20 | | * 2021-09-29 9724629 more detailed prints (Yangyao Chen)
21 | | * 2021-09-29 cf2dfa3 more detail prints (Yangyao Chen)
22 | | * 2021-09-29 4657f92 remove useless vars (Yangyao Chen)
23 | | * 2021-09-29 763eccb use new datatype converter (Yangyao Chen)
24 | | * 2021-09-29 cc6e66b move defs out-of-class (Yangyao Chen)
25 | | * 2021-09-29 42b71e3 add datatype initializers (Yangyao Chen)
26 | | * 2021-09-29 1c2ef07 init test for datatypes (Yangyao Chen)
27 | | * 2021-09-29 ea2f41b init the mpi test framework (Yangyao Chen)
28 | | * 2021-09-29 59936df move var-declare in-scope (Yangyao Chen)
29 | | * 2021-09-29 06985ef remove out-of-class default arguments (Yangyao Chen)
30 | | * 2021-09-29 6cfb3f3 remove long - not supported by extern32 (Yangyao Chen)
31 | | * 2021-09-29 f2c2a0f adopt datatype environment building and destroying (Yangyao Chen)
32 | | * 2021-09-29 301c4ee move definitions out-of-class (Yangyao Chen)
33 | | * 2021-09-29 17deb7a move definitions out-of-class (Yangyao Chen)
34 | | * 2021-09-29 f602dea use new typecv (Yangyao Chen)
35 | | * 2021-09-29 027eedd remove unused variables (Yangyao Chen)
36 | | * 2021-09-29 4e6aa0a add tests for env and datatypes (Yangyao Chen)
37 | | * 2021-09-29 3a9394c Force compiler flags (Yangyao Chen)
38 | | * 2021-09-29 1fdcceb Move implementations to inline (Yangyao Chen)
39 | | * 2021-09-23 fd5786b Remove a blank line (Yangyao Chen)
40 | | * 2021-09-23 6be32e6 Add a line number and barrier (Yangyao Chen)
41 | | * 2021-09-23 af7cdc5 Restructure the test data class (Yangyao Chen)
42 | | * 2021-09-23 d140eef Add tests for info method (Yangyao Chen)
43 | | * 2021-09-23 5ad7655 Reimplement the info method (Yangyao Chen)
44 | | * 2021-09-23 6a77fed Change the docs for Comm (Yangyao Chen)
45 | | * 2021-09-23 0fd7d18 Change the docs for Comm (Yangyao Chen)
46 | | * 2021-09-22 ff5d381 Change a title (Yangyao Chen)
47 | | * 2021-09-22 4e29bee Add contiguous buffer protocol (Yangyao Chen)
```

A E hipp --graph [-] 1:1 (0% 738L)

o: open split / O: open tab / gb: GBrowse / q: quit

wangk@Kosmos ➜

0:nvim\*

0 0 ▲ 0

① 12:26:55 2021-10-23