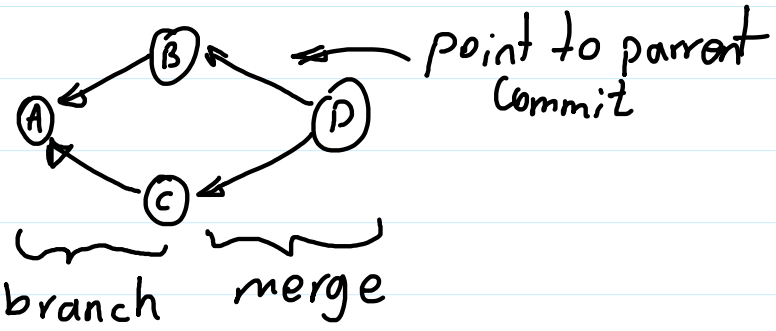
Git Week 2

Sunday, August 19, 2018 9:10 PM

Git graph model

- git models the relationship of commits with DAG
- You can see your project history as a graph

```
git log --oneline --graph
```



Git Objects:

1. **Commit**- A small text file
2. **Annotated tags**- A permanent reference to a commit
3. Tree- Directories and filenames in the project
4. Blob- The content of a file in the project

Git ID → the name of a git object

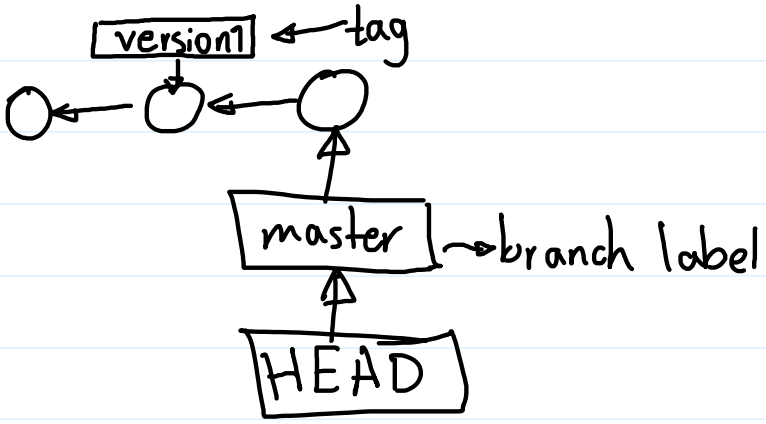
→ SHA-1
40 characters
or for short view only 7
or 10 characters

or 10 characters will show

- Use "git hash-object <file>" to create an SHA-1 for any content

Tag

Reference/Label attached to a specific commit.



- tag is used to indicate important commits
- Tags must be explicitly pushed to a remote repository
- A branch label is a reference to the tip of the branch

the tip of the branch

- HEAD is a reference that points to the current commit

- Reference is a user-friendy name that points to:
  - a commit SHA-1 hash
  - another reference
    known as symbolic reference
    HEAD -> master

git show HEAD
or SHA-1
↳ shows detail of the commit

- .git/refs/heads → heads → master
  stored refs ↳ tags ↳ v0.1

- One HEAD per repository

- .git/HEAD
  ↳ refs/heads/master

- Refers to a prior Commit
  ~ or ~1 = parrent

~ or ~1 = parrent

~2 or ~~ = parrent's parrent

git show HEAD

git show HEAD~1

git show HEAD~2

^^ first parent's first parent

Two Type of tag:

- lightwaight ~~→~~ not recommended
- Annotated ✓

git tag

↳ view all tags

git show v0.1

git tag <tagname> <commit>
lightweight HEAD

git tag -a | -m <msg> |
| -F <file> |

<tagname>

[<commit>] → default is HEAD

<tagname>
[<commit>] → default is HEAD

git push <remote> <tagname>
origin v0.1

git push <remote> --tags

Git Branches:

Benefits:

- fast & easy
- enable experimentation
- enable team development
- support multiple project version

lifetime:

- topic
  A feature, a bug fix, a hotfix
  a configuration change, etc

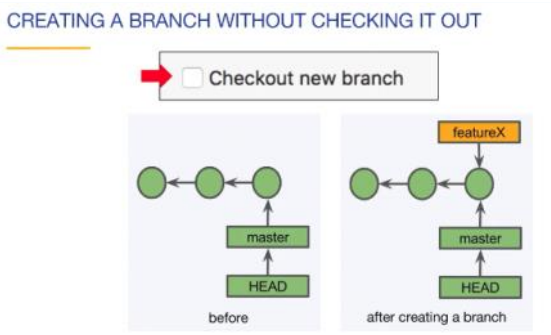
- long-lived
  master, develop, release, etc

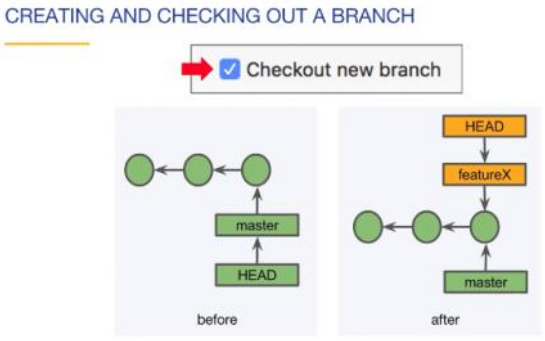
+ creating a branch creates a branch label

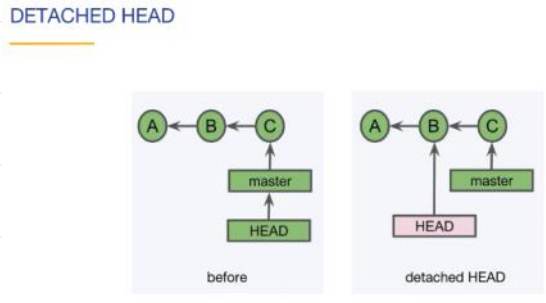
Checkout:

1- updates the HEAD reference

2- Updates the working tree with the commit's files







+ Dangling Commits will eventually be garbage collected

git branch -a ⇝ list of branches

git branch <branch-name>
git checkout <branch-name>
| git branch -b <branch-name> (only for new branches)

git branch -d <branch-name> ⇝ delete a branch
git branch -D <branch-name> ⇝ force the delete
be careful with this command

+ git reflog returns a local list of recent commits
  you can use it for recovering dangling commits and then use the git IDs to get back into the bussiness

```
git reflog

git branch -d <branch-name>
```

SHA-1 for the dangling commit

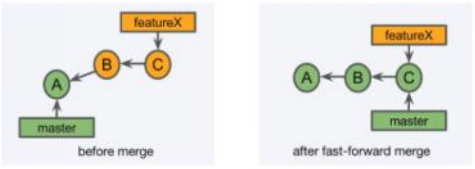
Merging:

Main types of merges:

1- Fast-forward merge
2- Merge commit
3- Squash merge
4- Rebase

fast-forward merge

Moves the base branch label to the tip of the topic branch



possible if no other commits have been made to the base branch since branching

Steps:

1- checkout master -> git checkout master

2- Merge branch -> git merge <branch>

a. attempting a fastforward merge ↳ git merge --no-ff <branch> is the default

3- Delete the branch -> git branch -d <branch>

Merge Commit

1- combines the commits at the tips of the merged branches

2- places the result in the merge commit

Quiz

1.
In Git, what is modeled as a directed acyclic graph?

- ( ) The staging area.
- ( ) The working tree.
- (•) The commit history.

2.
How are Git commits connected?

- ( ) A commit object contains the SHA-1 of its child or children.
- ( ) The staging area lists the connections.
- (•) A commit references its parent(s).

3.
What is a Git ID?

- ( ) The user's name and email address.
- ( ) The ID of the local repository.
- (•) The name of a Git object.

4.
If a large file changes by one character, what would you expect to happen to its corresponding SHA-1 value?

- ( ) It would not change.
- ( ) It would slightly change.
- (•) It would change drastically.

5.
What do branch labels point to?

- (•) The most recent commit of a branch.
- ( ) The initial commit of a branch.
- ( ) Every commit of a branch.

6.
How many HEAD references are in a local repository?

- ( ) One for each commit.
- ( ) One for each branch label.
- (•) One.

7.
Which one of these statements is correct?

- ( ) A tag is another name for a branch label.
- (•) A tag always points to a specific commit.
- ( ) The HEAD reference always points to a tag.

8.
What happens when a branch is created?

- ( ) Commits are copied.
- ( ) The HEAD reference changes.
- (•) A branch label is created.

9.
Which one of these statements is correct?

- ○ Checkout retrieves content from the remote repository.
- ◉ Checkout updates the working tree and HEAD reference.
- ○ Checkout prevents others from changing a branch.

10.
What does a detached HEAD mean?

- ○ The HEAD reference points to a branch label.
- ○ The HEAD reference does not point to anything.
- ◉ The HEAD reference points directly to a commit SHA-1.

11.
What does "deleting a branch" immediately do?

- ◉ Deletes a branch label.
- ○ Deletes only the commits that are unique to the branch.
- ○ Deletes all of the commits of the branch.

12.
Which one of the following statements is true?

- ○ A commit can only belong to one branch at a time.
- ◉ Merging combines the work of branches.
- ○ A merge always creates a new commit.

13.
Which one of the following statements about fast-forward merges is true?

- ◉ The merge moves a branch label.
- ○ The merge may result in a merge conflict.
- ○ The merge may change some commits.

14.
If Git informs you that a fast-forward merge is not possible, which one of these statements is probably true?

- ○ The merge has merge conflicts.
- ◉ A commit was made on the base branch after the topic branch was created.
- ○ The checked out commit has multiple parents.

15.
Which one of these statements about a merge involving a merge commit is true?

- ○ A merge commit results in a linear commit history.
- ◉ Git places the result of the merge into a new commit.
- ○ The merge is aborted if there are merge conflicts.