

- [Sign in](#)
- [Tips&Tricks](#)
- [Courses](#)
- [Test Projects](#)
- [Forum](#)
- [Sign up](#)

Quiz result: 87.5%

You may copy this URL to reference the quiz result later:

<https://modernweb.design/quizzes/jYd1WkJPzws6r1tMWzYAZp8P/result>

Attendent Name: NG CHENG CHUN.

1: When creating a branch in git, what happened behind the scene? [git003]

1. ☐ The git system creates a new pointer in remote server and syncs the new branch to local.
2. ☒ The git system creates a new pointer that points to current commit.
3. ☐ The git system clones the current commit into the new branch.
4. ☐ The git system copies the current project folder as a new branch.

2: Usually what files we want to ignore in version control system? [git004]

1. ☒ Files that contain secret tokens, personal preferences, editor settings.
2. ☐ We should track all files in git version control system.
3. ☐ Files that contain personal preferences, application configurations.
4. ☐ Files that are binary, such as executables, images, videos.

3: How we can track the database configurations in git versions control but not storing the database passwords in git repository? [git005]

1. ☐ We may encrypt the database configuration file when storing in remote repository.
2. ☐ We could ignore the whole database configuration file.
3. ☒ We could store the passwords in environment variables without directly writing the passwords in the configuration file.
4. ☐ We could manually input the password in the configuration file in production server.

4: How to switch branch in git version control system? [git006]

1. ☐ By using ``git switch <branch_name>``.

2. ☒ By using ``git checkout <branch_name>``.
3. ☐ By using ``git branch <branch_name>``.
4. ☐ By using ``git -b <branch_name>``.

5: Git tracks files but not folder. Is there any way to keep an empty folder in the repository? [git007]

1. ☐ By putting the empty folder under the project folder.
2. ☐ We may avoid tracking empty folder. Why we need to track empty folder?
3. ☒ By creating a blank hidden file in the target folder.
4. ☐ By setting a config in git to track folders.

6: When doing ``git log``, how to show a readable branch path in command line to have a big picture on the history? [git009]

1. ☒ `git log --oneline --graph`
2. ☐ `git log --oneline`
3. ☐ `git log --decorate`
4. ☐ `git log --graph`

7: What is the usage of ``git diff --cached``? [git011]

1. ☐ It diffs the cached changes to the master branch.
2. ☒ It diffs the cached changes to latest in-staged changes in current folder.
3. ☐ It diffs the staged changes to committed code in current branch.
4. ☐ It diffs the latest un-staged changes to committed code in current branch.

8: ``git status -s`` shows a short version of the status. When to use the long form of ``git status`` [git010]

1. ☐ The short form cannot show if the files are changed, staged, or committed.
2. ☒ The long form shows a detail diff of the changes, which is easy to know what will be committed.
3. ☐ The long form shows a summary on the current status, including instruction suggestions for further operations.
4. ☐ The short form cannot show if any files are deleted and not committed.

9: Why we don't want to always use ``git commit -am``? [git012]

1. ☐ Separating ``git add`` and ``git commit`` allows committing all file changes into the control system.
2. ☐ ``git commit -am`` allows committing changes without messages.
3. ☒ Separating ``git add`` and ``git commit`` allows developers to carefully review what to be committed.

- ☐ `git commit -am` is fast and one line operation to quickly commit current changes.

10: Why sometimes we want to add part of a file into the control system? [git013]

- ☐ We want to commit each developer's work into individual commit.
- ☒ We want to separate each commit to include only related code changes.
- ☐ We want to make each commit as small as possible.
- ☐ We want to commit each function block into individual commit.

11: Where is git control system stores the project repository for each working project? [git014]

- ☐ It stores the repo in home directory.
- ☐ It stores the repo in remote server.
- ☐ It stores the repo in git's application folder.
- ☒ It stores the repo in `.git` folder in each working project folder.

12: What makes a good commit message? [git017]

- ☐ Begins with verb.
- ☐ Use first line as summary.
- ☐ Use present tense.
- ☒ All of the choices.

13: How to mark changes from un-tracked status to staging? [git020]

- ☐ `git track <filename>`
- ☒ `git add <filename>`
- ☐ `git commit <filename>`
- ☐ `git stage <filename>`

14: How to mark changes from staging status to a new commit? [git021]

- ☐ `git commit -am <message>`
- ☒ `git commit -m <message>`
- ☐ `git commit`
- ☐ `git commit <message>`

15: In what scenarios we want to create a new branch? [git022]

- ☐ When we want to create a new feature.

2. ☐ When we want to fix an issue in code.
3. ☐ When we want to mark a stable version for production deployment.
4. ☒ All of the choices.

16: Where we can store the git repository remotely? [git023]

1. ☐ Through repository service provider, such as Github, Bitbucket.
2. ☐ By storing a private repository in a private server through SSH.
3. ☐ By storing the repository in a shared folder with a remote file server.
4. ☒ All of the choices.

Please login to chat.

© 2016—2017, [Makzan](#) | [About](#) | [Looking for the original Ruby on Rails course note?](#)