Git Cheat Sheet

Prepare environment

Use global config
```
$ git config --global <key> <value>
```

Deal with CR/LF issues (Windows)
```
$ git config core.autocrlf true
```

Set name
```
$ git config user.name <name>
```

Set email (in case of SSH issues)
```
$ git config user.email <email>
```

Set alias
```
$ git config alias.<alias> <cmd>
```

Common aliases

st status
br branch
ci commit
co checkout

Create a repository

Create new local repository
```
$ git init <project_name>
```

Download from an existing repository
```
$ git clone <my_url>
```

Observe your repository

List new or modified files
```
$ git status
```

Show the changes to files not yet staged / staged
```
$ git diff [--cached]
```

Show all changes
```
$ git diff HEAD
```

Show the changes between two commits
```
$ git diff <commit_id1> <commit_id2>
```

List che change dates and authors to a file
```
$ git blame <file>
```

Show the file changes for a commit id and/or file
```
$ git show <commit>:<file>
```

Show full change history
```
$ git log
```

Show change history for a file/directory including diffs
```
$ git log -p <file/directory>
```

Show modify/delete conflict – deleted by me
```
$ git diff ...origin/master – <file>
```

Show modify/delete conflict – deleted by them
```
$ git diff origin/master...  – <file>
```

Working with branches

List all local (local and remote) branches
```
$ git branch [-av]
```

Switch to a different branch
```
$ git checkout <branch_name>
```

Switch to a new branch
```
$ git checkout -b <branch_name>
```

Create a new branch
```
$ git branch <branch_name>
```

Delete a branch
```
$ git branch -d <branch_name>
```

Delete a remote branch
```
$ git push <origin> --delete
<branch_name>
```

Merge a branch into the current branch
```
$ git merge <branch_name>
```

Merge a branch into the current branch without using fast-forward
```
$ git merge --no-ff <branch_name>
```

Tag the current branch
```
$ git tag <tag>
```

Make a change

Stage a file, ready for commit
```
$ git add <file>
```

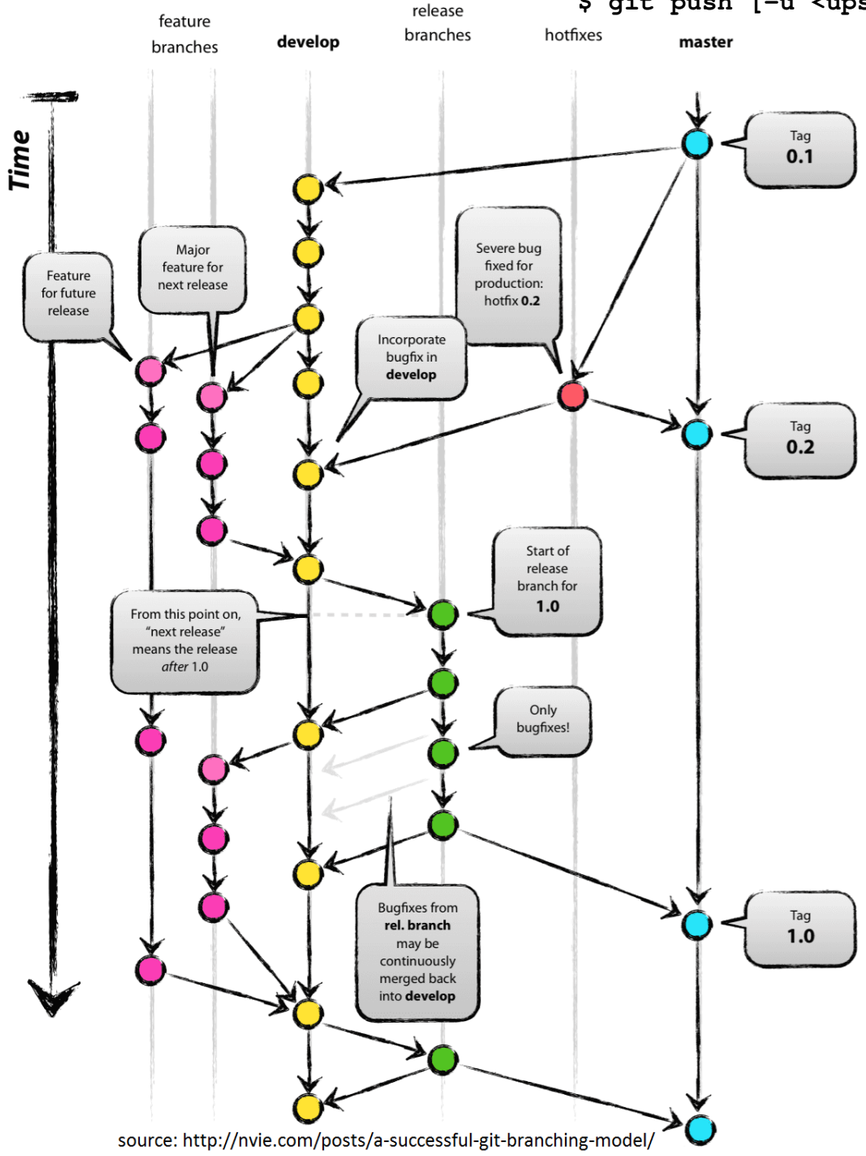
Add all files
```
$ git add --all
```

Commit all staged (tracked) files
```
$ git commit [-a] -m "commit message"
```

Change last commit
```
$ git commit --amend
```

Unstage file, keeping the file changes
```
$ git reset <file>
```

Revert everything to the last commit
```
$ git reset --hard
```



source: http://nvie.com/posts/a-successful-git-branching-model/

Synchronize

Get the latest changes without merge (all branches)
```
$ git fetch [--all]
```

Fetch the latest changes and merge
```
$ git pull
```

Fetch the latest changes and rebase
```
$ git pull -- rebase
```

Push local changes (and set upstream for later use)
```
$ git push [-u <upstream> <branch>]
```

Stash

Save local changes into stash
```
$ git stash
```

Apply changes from stash
```
$ git stash apply
```

Save local changes as a new stash
```
$ git stash save
```

List all the stashes
```
$ git stash list
```

Clears all the stashes
```
$ git stash clear
```