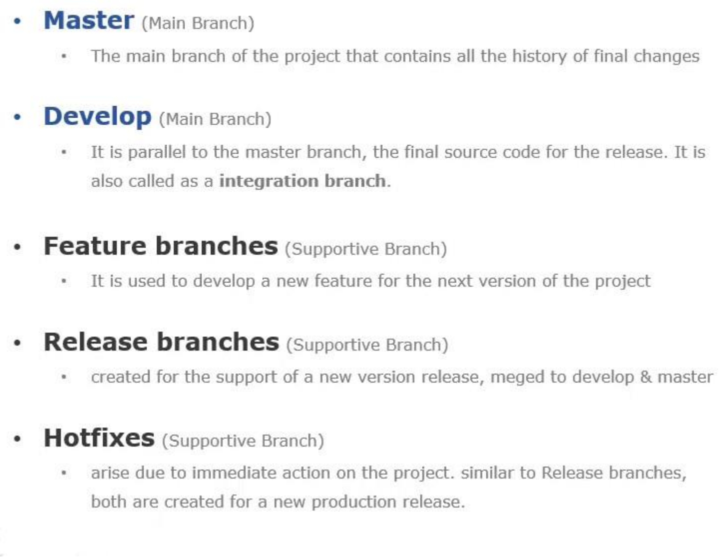
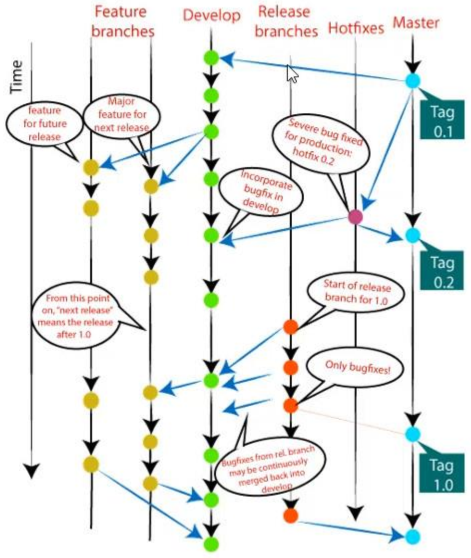
Best Git Cheat Sheet
For Web Developers

```
Git config  --global user.name "siavash aghazadeh"
Git config --global user.email siavash.agh@gmail.com
Git init      ,   Git status
Git add .     ||    add <file>
Git add patch <file>
Git log --oneline --all --graph
Git commit      ||      Git commit -m "message"
Git show <here hash of an commit>
Git restore –staged <file>
Git restore <file>
Git restore --sourse <commit hash> <file>
Git reset  HEAD~2      ||  Git reset <hash of an commit>
Git branch
Git branch <branch name>
branch <branch name>(<hash of commit>||<HEAD~n>)
```



- **Master** (Main Branch)
  - The main branch of the project that contains all the history of final changes
- **Develop** (Main Branch)
  - It is parallel to the master branch, the final source code for the release. It is also called as a **integration branch**.
- **Feature branches** (Supportive Branch)
  - It is used to develop a new feature for the next version of the project
- **Release branches** (Supportive Branch)
  - created for the support of a new version release, meged to develop & master
- **Hotfixes** (Supportive Branch)
  - arise due to immediate action on the project. similar to Release branches, both are created for a new production release.


Feature branches
Develop
Release branches
Hotfixes
Master
Time
feature for future release
Major feature for next release
Tag 0.1
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Tag 1.0
Bugfixes from rel. branch may be continuously merged back into develop

```
Git switch (switch between branchs)
Git checkout (switch between branchs)
(get to master branch and..) Git merge <branch name>
Git diff <branch name>--<branch name>
Git stash     ,    Git stash -u   <= (this for untracked files)
Git stash list
Git stash show <stash id>
Git stash (apply || pop) <stash id>
Git stash clear
Git remote
Git remote add <name> <url>
Git remote rm <name>
Git remote rename <old-name> <new-name>
Git remote show <name>
Git push <remote-name> <branch-name>
Git clone <url from github>
Git pull <remote-name>
```