Cheatography

git-flow Cheat Sheet

by glisignoli via cheatography.com/20258/cs/3083/

Git workflows

Start a **feature**, work on it, **finish** it.

*Git flow merges **feature** back into **develop***

Work on **develop**, start a **release**, **finish** it

*Git flow merges **release** into **master** and **develop***

Start a **hotfix**, work on it, **finish** it.

*Git flow merges **hotfix** into **master** and **develop***

Initialise a new repo

Normal

```
$ mkgitrepo -n name
```

Puppet

```
$ mkgitrepo -p -n name
```

Normal will create your remote repo in `/opt/git/name`

Puppet will create your remote repo in `/opt/g it/ pup pet /mo dules/name`

Clone an existing repository

Normal

```
$ git clone git@pv iaa sfu jno v02 :/o pt/git/name.git
```

Puppet

```
$ git clone git@pv iaa sfu jno v02 :/p upp et/mo dul es/name.git
```

Locations for cloning repositories or modules

Branch checking

```
$ git branch -a
* develop
  master
  remote s/o rig in/ develop
  remote s/o rig in/ master
```

Shows the current remote a local braches

Switching Branches

```
git checkout branchname
```

Switches to a new branch. In git flow you will want to be working on the develop branch or features branch.

Checking git transactions

Command Line

```
$ git log
```

Terminal Interface

```
$ tig
```

Status Checking

```
$ git status
```

Shows the status of the current directory

Commiting

Adding new files

```
$ git add *
```

Committing changes

```
$ git commit -a -m " Change descri pti on"
```

`git commit -a` will only commit already tracked files. If you create new files you need to use `git add *`

Git flow Features branch

Creating Features

```
$ git flow feature start " Fea tur eNa me"
```

Publish feature to remote

```
$ git flow feature publish " Fea tur eNa me"
```

Get a published feature

```
$ git flow feature pull origin " Fea tur eNa me"
```

Finishing a Feature

```
$ git flow feature finish " Fea tur eNa me"
```

Features are useful for large repositories with multiple files, or repositories where multiple people will be working at the same time.

Git flow Hotfix branch

Creating a Hotfix

```
$ git flow Hotfix start " Hot fix
```

Publish Hotfix to remote

```
$ git flow Hotfix publish " Hot fi
```

Get a published hotfix

```
$ git flow hotfix pull origin " Ho m e"
```

Finishing a Hotfix

```
$ git flow hotfix finish " Hot fix
```

The hotfix branch is used to create fixes to the master branch, and then merge them back into the develop branch.

Git flow Release branch

Creating a Release

```
$ git flow Release start " Rel eas
```

Publish Release to remote

```
$ git flow Release publish " Rel e
```

Get a published release

```
$ git flow release pull origin " R me"
```

Finishing a Release

```
$ git flow release finish " Rel ea
```

The release branch is used to merge the *develop* branch into *master*

Pushing changes

```
$ git push --all
```

Pushes your changes to the remote repository