CHEATSHEET FOR

Git tricks

Refs

```
HEAD^       # 1 commit before head
HEAD^^      # 2 commits before head
HEAD~5      # 5 commits before head
```

Branches

```
# create a new branch
  git checkout -b $branchname
  git push origin $branchname --set-upstream

# get a remote branch
  git fetch origin
  git checkout --track origin/$branchname

# delete local remote-tracking branches (lol)
  git remote prune origin

# list merged branches
  git branch -a --merged

# delete remote branch
  git push origin :$branchname

# go back to previous branch
  git checkout -
```

Collaboration

```
# Rebase your changes on top of the remote master
  git pull --rebase upstream master

# Squash multiple commits into one for a cleaner git log
# (on the following screen change the word pick to either
  git rebase -i $commit_ref
```

Submodules

```
# Import .gitmodules
  git submodule init

# Clone missing submodules, and checkout commits
  git submodule update --init --recursive

# Update remote URLs in .gitmodules
# (Use when you changed remotes in submodules)
  git submodule sync
```

Diff

Diff with stats

```
git diff --stat
app/a.txt     | 2 ++
app/b.txt     | 8 +++---
2 files changed, 10 insertions(+), 84 deletions(-)
```

Just filenames

```
git diff --summary
```

Log options

```
--oneline
  e11e9f9 Commit message here

--decorate
  shows "(origin/master)"

--graph
  shows graph lines

--date=relative
  "2 hours ago"
```

Misc

Cherry pick

```
git rebase 76acada^
```

Misc

```
# get current sha1 (?)
  git show-ref HEAD -s

# show single commit info
  git log -1 f5a960b5

# Go back up to root directory
  cd "$(git rev-parse --show-top-level)"
```

Short log

```
$ git shortlog
$ git shortlog HEAD~20..   # last 20 commits

James Dean (1):
    Commit here
    Commit there

Frank Sinatra (5):
    Another commit
    This other commit
```

Bisect

```
git bisect start HEAD HEAD~6
git bisect run npm test
git checkout refs/bisect/bad   # this is where it screwed
git bisect reset
```

Manual bisection

```
git bisect start
git bisect good  # current version is good

git checkout HEAD~8
npm test         # see if it's good
git bisect bad   # current version is bad

git bisect reset  # abort
```

Searching

```
git log --grep="fixes things"  # search in commit messages
git log -S"window.alert"       # search in code
git log -G"foo.*"              # search in code (regex)
```

GPG Signing

```
git config set user.signingkey <GPG KEY ID>       # Sets G

git commit -m "Implement feature Y" --gpg-sign    # Or -S,

git config set commit.gpgsign true                # Sign c
git commit -m "Implement feature Y" --no-gpg-sign # Do not
```