**User command sequence for an existing repo.**

**Cloning the git repo   :**

git clone 'url'

**Create a New branch  :**

git branch  branchname

**List all branches  :**

git branch –a

**Switch Branch**

git checkout branchname

**Check changes in the local git repo**

git status

**Adding all the untracked files**

git add \*

**Commiting the changes in the file**

git commit -m “comment” filename

**For getting the latest codebase from server git repo  to local git repo**

git pull

**Pushing changes to git repo**

git push origin branchname

**Deleting the Branch**

git delete –d branchname

**Merging a branch to current branch**

git merge branchname

**Drush Commands:**

**Import Databse into server**

drush sql-cli < database\_file.sql

**Download Module**

drush dl module name (eg: drush dl webform)

**Enable module**

drush en module name (eg: drush en webform)

**Disable Module**

drush dis module name (eg: drush dis webform)

**Uninstall Module**

drush pm-uninstall module name (eg: drush dis webform)

**Update a module**

drush pm-update module name (eg: drush dis webform)

**List all modules and themes :**

drush pm-list

**List only modules :**

drush pm-list --type=Module

**List all modules except core modules :**

drush pm-list --type=Module --no-core

**List all enabled modules except those from the core :**

drush pm-list --type=Module --no core -- status=enabled

**Get status about your site**

drush status

**Save database backup of a site**

drush  sql-dump > D:/filename.sql

**Clear the cache** drush cc all

**Clear the cache, even if Drupal is broken** drush sql-query "DELETE FROM cache"

sql-query executes sql queries in the database where Drupal is installed.

**Disable a module, even if Drupal is broken**

drush sql-query "UPDATE system SET status = '0' WHERE name = 'module\_name'"

drush sql-query "DELETE FROM cache\_bootstrap WHERE cid = 'system\_list'"

**See if a module is available** drush pm-list | grep module\_name

**See if a module is enabled** drush pm-list --status=enabled | grep module\_name

**See all contrib and custom modules that are enabled** drush pm-list --type=Module --no-core --status=enabled

**Update a module** drush up module\_name

**Update Drupal** drush up drupal

**Update all contrib modules and Drupal**  
 drush up

**Run update.php** drush updb

**Send an e-mail** drush php-eval "print mail('pere@example.com', 'Subject', 'Message', 'From: pere@example.com');"

**Delete a field** drush field-delete fieldname

**Delete an instance of a field** drush field-delete fieldname --bundle=article

**Manually delete a field and all its data (not recommended)**

drush sql-query "DELETE FROM field\_config WHERE field\_name = 'fieldname'"

drush sql-query "DELETE FROM field\_config\_instance WHERE field\_name =' fieldname'"

drush sql-query "DROP TABLE field\_data\_field\_fieldname"

drush sql-query "DROP TABLE field\_revision\_field\_fieldname"

**Set a password for a user** drush upwd --password="asdf" admin

**Block a user** drush user-block joe

**Log all users out** drush sql-query 'TRUNCATE TABLE sessions;'

**Log out a specific user** drush sql-query 'DELETE FROM sessions WHERE uid = 2;'

**Get a one-time login link for the Administrator user** drush uli

**Run cron** drush cron

**Run Ultimate Cron** drush cron-run

**Set a variable** drush vset variable\_name 'value'

**Get the value of a variable** drush vget variable\_name

**See last logged events (watchdog)** drush ws

**See logged events in real time** drush ws --tail

**Run a php script with Drupal bootstrapped**

drush scr --uri=domain.com script.php

Use --uri to specify the host name. Use --root to specify the site directory (in cron, for example):

/usr/bin/drush --root=/var/www/drupal scr /var/www/drupal/sites/all/scripts/example.php >>/root/logs/scripts/example\_log.txt 2>>/root/logs/scripts/example\_errors.txt

**Run custom code** drush eval "variable\_set('foo', 'bar');"

**Repopulate database tables used by menu functions** drush eval "menu\_rebuild();"

See https://api.drupal.org/api/drupal/includes!menu.inc/function/menu\_rebuild/7

**Rescan all code in modules or includes directories, storing the location of each interface or class in the database**

drush eval "registry\_rebuild();"

Useful when moving installed modules. Seehttp://drupal.stackexchange.com/q/17657/10086

If does not work because Drupal cannot bootstrap, do it that way:

drush dl registry\_rebuild

drush rr

**Update a Feature with database changes**

drush fu feature\_name

**Revert a Feature, update the database to match the code**

drush fr feature\_name

**Revert all features**

drush features-revert-all -y

Change error level

**Show only errors and warnings**

drush vset -y 1

**Show all**

drush vset -y 2

**Hide everything**

drush vset -y 0

**Enable maintenance mode**

drush vset -y maintenance\_mode 1

**Open a MySQL console logged in**

drush sql-cli

**Import a backup of the database**

drush sql-cli < dump.sql

**Export a backup of the database**

drush sql-dump | gzip --stdout > $(date +\%Y-\%M-\%d-\%H-\%M-\%S).sql.gz

**Remove all database tables (empty the database)**

drush sql-drop

Useful before importing a database backup.

**See all drush aliases**

drush site-alias

**Use an alias**

drush @aliasname command

**Flush image styles**

drush image-flush all

**Install Drupal**

drush site-install

**Delete all content of specific content types**

sudo drush generate-content 0 --kill --types=facebook,instagram,tweet

Note that you need Devel and Devel Generate modules. Alternatives are drush migrate-wipe and Delete all module.

**Delete content type**

drush php-eval "node\_type\_delete('job\_posting')node\_types\_rebuild();menu\_rebuild();"

**Generate random content**

sudo drush generate-content 20 --types=page

**Create a boilerplate for a new module**

drush mb my\_module menu cron --write --name="My module"

**Run sanitization operations on the current, non-production database**

drush sql-sanitize

You may implement hook\_sql\_sync\_sanitize() in your custom modules to clean specific data.

Same as above, but also setting a default login password for all users

drush sql-sanitize --sanitize-password='newpassword'

Same as above, but also sanitizing all user emails

drush sql-sanitize --sanitize-password='newpassword' --sanitize-email

**Run Drush commands for a specific site**

drush -l www.gonyon.com en rules

**Drush for MultiSite**

**For normal commands like download,enabled,disable,list modulesyou can add @sites before command in your root directory of multisite**

Ex drush @sites dl views -y

***For Taking backup you have to run below command,this will save back in C:\Users\username\drush-backup***

drush @sites sql-dump --result-file