CREATE "The database or its objects are created with this command (like table, index, function, views, store procedure, and triggers). There are two types of CREATE statements in SQL, one is for the creation of a database and the other for a table.

DROP TABLE table\_name Delete a whole table

DROP DATABASE db\_name The DROP command can be used to delete a whole database

ALTER In an existing table, this command is used to add, delete/drop, or edit columns.

TRUNCATE Used to indicate the table’s extents for deallocation (empty for reuse). This procedure removes all data from a table quickly, usually circumventing a number of integrity checking processes. It was included in the SQL:2008 standard for the first time. It is somewhat equivalent to the delete command.

INSERT It is used to insert data into a table’s row.

UPDATE In SQL, the UPDATE statement is used to update data in an existing database table. We can use the Alter statement to update single or several columns depending on our needs.

DELETE In SQL, the DELETE statement is used to delete records from a table. Depending on the condition we set in the WHERE clause, we can delete a single record or numerous records.

GRANT User access privileges to a database are given by this command. It can be used to grant SELECT, INSERT, UPDATE, and DELETE privileges to a user on a single table or several tables.

REVOKE To take back permissions from the user REVOKE command is used. It is used to revoke a privilege (by default) or a specific command, such as UPDATE or DELETE, depending on the situation.

COMMIT To save all the transactions in the database Commit is used.

ROLLBACK All modifications must be canceled if any of the SQL grouped statements produce an error. The term “rollback” refers to the process of undoing changes. This command can only be used to reverse transactions that have occurred since the last COMMIT or ROLLBACK command.

SAVEPOINT It’s used to roll back a transaction to a specific point rather than the complete transaction.

pip install package-nameExample: pip is a package manager which is written in Python language. It is used to install and manage software packages. The pip install command is used to install any software package from an online repository of public packages, called the Python Package Index. To run this command in windows you need to open your Windows PowerShell and then use the following syntax to install any package.

print(object)Example: print command is used to print a message on the screen or other standard output device. The message can be a string or any other object. The print command can be used to print any type of object like integer, string, list, tuple, etc.

type(object)Example: type command is used to check the type or class of an object.

range(start, stop, step) range command is used to generate a sequence of integers starting from 0 by default to n where n is not included in the generated numbers. We use this command in for loops mostly.

round(number, digits) round command is used to round a number to a given precision in decimal digits. That means if you have so many digits after decimal in a float point number then you can use the round command to round off the specified number. You can mention how many digits you want after the decimal point.

input(message) input command is used to take input from the user. The flow of the program will be stopped until the user has entered any value. Whatever the user enters it will be converted into a string by the input function. If you want to take an integer as input then you have to convert it explicitly.

len(object) len command or len() function is used to get the number of items in an object. If the object is a string then len() function returns the number of characters present in it. If the object is a list or tuple it will return the number of elements present in that list or tuple. len() gives an error if you try to pass an integer value to it.

stringname.isalnum() isalnum(): It checks whether all the characters of a given string are alphanumeric or not. It returns a boolean value.

stringname.capitalize() capitalize() function changes the first character of the string to uppercase if it’s lowercase. If the first character is uppercase or an integer or any special character, it doesn’t do anything.

string.find(substring) find() command is used to search for a substring in a string. It returns the index of the first occurrence of the substring if it is present otherwise it returns -1.

stringname.count(substring, start, end) Example:

string.center(length, character) center command is used to align a string in the center, using a specified character (by default it is space) as the fill character.

list.append(element) append(): append command is used to add an element at the end of the list.

list.copy() copy() is used to create a new copy of the list object. It returns a new list object.

listname.insert(position, element) insert command is used to add an element at a specified position in the list object. It takes two parameters position and element.

listname.pop(position) pop() method is used to remove an element from a specified position in the list. It returns the element after removing it from the list.

list.reverse() reverse method reverses the order of all elements in the list. It modifies the original list object and it doesn’t return anything.

list.sort() sort method is used to sort the elements of the list in ascending order by default.

tuple.count(element) count(): count method is used to count the occurrences of an element in the tuple.

tuple.index(element) This method is used to find the index of the first occurrence of an element. If the element is not found in the whole tuple then it will raise a ValueError.

setname.add(element) add(): add command is used to add a new element in the set.

setname.clear() clear removes all the elements of a set. It does not take any parameters.

setname.discard(element) discard is used to remove the specified element from the set. If the specified element is not found in the set it will not give an error.

setname.remove(element) remove command is also used to remove a specified element from the set but it is different from discard because remove will give an error if the specified element is not found in the set.

setA.difference(setB) difference method is used to get a set that contains the difference of two sets. The difference of sets means it will have only those elements that are present in only one set and not in another set. Suppose we have two sets A and B. Set A has {1,2,3} and set B has {2, 4, 6}. Then the difference of A and B will be {1,3}.

setA.difference\_update(setB) difference\_update method is used to get a set of elements that are present in the first set and not common in both sets. That means it removes the elements that exist in both sets. It does not return a new set, it just removes common elements from the first set.

set.intersection(set1, set2, … setn) intersection method returns a set having elements that exist in all the specified sets.

setA.issubset(setB) issubset method checks whether all elements of set A are present in set B or not. It returns a boolean value.

setA.symmetric\_difference(setB) This method returns a set containing elements from both the sets except those that are common in both sets.

setA.union(setB) union method returns a set containing all elements from both sets, except the duplicated ones.

dict.fromkeys(keys, value) fromkeys(): fromkeys() method is used to generate a dictionary with specified keys and a specified value.

dictionary.get(key, value) get method is used to get a value of the specified key. If a key is not found in the dictionary it will return nothing unless we specify something in the parameters.

dictionary.items() items method is used to display the dictionary elements. It will display all the key-value pairs present in the dictionary. It returns a view object which will contain all the key-value pairs as tuples in a list. It does not take any parameters.

dictionary.keys() keys method is used to get all the keys present in the dictionary. It returns a view object containing all keys of the dictionary as a list. It does not take any parameters.

dictionary.values() values method is used to get all the values present in the dictionary. It returns a view object containing all values of the dictionary as a list. It does not take any parameters.

dictionary.pop(key) pop method is used to remove a key-value pair from the dictionary by specifying the key. It returns the value of the key-value pair that is need to be removed.

dictionary.popitem() popitem command is used to remove the last inserted pair from the dictionary. It does not take any parameters. It returns the removed pair as a tuple.

dictionary.setdefault(key, value) setdefault method is used to get the value of a specified key. If the key does not exist it will insert the key with the value passed as a parameter. If you don’t specify any value it will insert the key with value None.

Command Description

git-add Add file contents to the index

git-am Apply a series of patches from a mailbox

git-archive Create an archive of files from a named tree

git-bisect Use binary search to find the commit that introduced a bug

git-branch List, create, or delete branches

git-bundle Move objects and refs by archive

git-checkout Switch branches or restore working tree files

git-cherry-pick Apply the changes introduced by some existing commits

git-citool Graphical alternative to git-commit

git-clean Remove untracked files from the working tree

git-clone Clone a repository into a new directory

git-commit Record changes to the repository

git-describe Give an object a human readable name based on an available ref

git-diff Show changes between commits, commit and working tree, etc

git-fetch Download objects and refs from another repository

git-format-patch Prepare patches for e-mail submission

git-gc Cleanup unnecessary files and optimize the local repository

git-grep Print lines matching a pattern

git-gui A portable graphical interface to Git

git-init Create an empty Git repository or reinitialize an existing one

git-log Show commit logs

git-maintenance Run tasks to optimize Git repository data

git-merge Join two or more development histories together

git-mv Move or rename a file, a directory, or a symlink

git-notes Add or inspect object notes

git-pull Fetch from and integrate with another repository or a local branch

git-push Update remote refs along with associated objects

git-range-diff Compare two commit ranges (e.g. two versions of a branch)

git-rebase Reapply commits on top of another base tip

git-reset Reset current HEAD to the specified state

git-restore Restore working tree files

git-revert Revert some existing commits

git-rm Remove files from the working tree and from the index

git-shortlog Summarize git log output

git-show Show various types of objects

git-sparse-checkout Reduce your working tree to a subset of tracked files

git-stash Stash the changes in a dirty working directory away

git-status Show the working tree status

git-submodule Initialize, update or inspect submodules

git-switch Switch branches

git-tag Create, list, delete or verify a tag object signed with GPG

git-worktree Manage multiple working trees

gitk The Git repository browser

scalar A tool for managing large Git repositories

git-config Get and set repository or global options

git-fast-export Git data exporter

git-fast-import Backend for fast Git data importers

git-filter-branch Rewrite branches

git-mergetool Run merge conflict resolution tools to resolve merge conflicts

git-pack-refs Pack heads and tags for efficient repository access

git-prune Prune all unreachable objects from the object database

git-reflog Manage reflog information

git-refs Low-level access to refs

git-remote Manage set of tracked repositories

git-repack Pack unpacked objects in a repository

git-replace Create, list, delete refs to replace objects Interrogators:

git-annotate Annotate file lines with commit information

git-blame Show what revision and author last modified each line of a file

git-bugreport Collect information for user to file a bug report

git-count-objects Count unpacked number of objects and their disk consumption

git-diagnose Generate a zip archive of diagnostic information

git-difftool SHow changes using common diff tools

git-fsck Verifies the connectivity and validity of the objects in the database

git-help Display help information about Git

git-instaweb Instantly browse your working repository in gitweb

git-merge-tree Perform merge without touching index or working tree

git-rerere Reuse recorded resolution of conflicted merges

git-show-branch Show branches and their commits

git-verify-commit Check the GPG signature of commits

git-verify-tag Check the GPG signature of tags

git-version Display version information about Git

git-whatchanged Show logs with differences each commit introduces

gitweb Git web interface (web frontend to Git repositories)

git-archimport Import a GNU Arch repository into Git

git-cvsexportcommit Export a single commit to a CVS checkout

git-cvsimport Salvage your data out of another SCM people love to hate

git-cvsserver A CVS server emulator for Git

git-imap-send Send a collection of patches from stdin to an IMAP folder

git-p4 Import from and submit to Perforce repositories

git-quiltimport Applies a quilt patchset onto the current branch

git-request-pull Generates a summary of pending changes

git-send-email Send a collection of patches as emails

git-svn Bidirectional operation between a Subversion repository and Git

git-revert is about making a new commit that reverts the changes made by other commits.

git-restore is about restoring files in the working tree from either the index or another commit. This command does not update your branch. The command can also be used to restore files in the index from another commit.

git-reset is about updating your branch, moving the tip in order to add or remove commits from the branch. This operation changes the commit history.

git-apply Apply a patch to files and/or to the index

git-checkout-index Copy files from the index to the working tree

git-commit-graph Write and verify Git commit-graph files

git-commit-tree Create a new commit object

git-hash-object Compute object ID and optionally create an object from a file

git-index-pack Build pack index file for an existing packed archive

git-merge-file Run a three-way file merge

git-merge-index Run a merge for files needing merging

git-mktag Creates a tag object with extra validation

git-mktree Build a tree-object from ls-tree formatted text

git-multi-pack-index Write and verify multi-pack-indexes

git-pack-objects Create a packed archive of objects

git-prune-packed Remove extra objects that are already in pack files

git-read-tree Reads tree information into the index

git-replay EXPERIMENTAL: Replay commits on a new base, works with bare repos too

git-symbolic-ref Read, modify and delete symbolic refs

git-unpack-objects Unpack objects from a packed archive

git-update-index Register file contents in the working tree to the index

git-update-ref Update the object name stored in a ref safely

git-write-tree Create a tree object from the current index

git-cat-file Provide contents or details of repository objects

git-cherry Find commits yet to be applied to upstream

git-diff-files Compares files in the working tree and the index

git-diff-index Compare a tree to the working tree or index

git-diff-tree Compares the content and mode of blobs found via two tree objects

git-for-each-ref Output information on each ref

git-for-each-repo Run a Git command on a list of repositories

git-get-tar-commit-id Extract commit ID from an archive created using git-archive

git-ls-files Show information about files in the index and the working tree

git-ls-remote List references in a remote repository

git-ls-tree List the contents of a tree object

git-merge-base Find as good common ancestors as possible for a merge

git-name-rev Find symbolic names for given revs

git-pack-redundant Find redundant pack files

git-rev-list Lists commit objects in reverse chronological order

git-rev-parse Pick out and massage parameters

git-show-index Show packed archive index

git-show-ref List references in a local repository

git-unpack-file Creates a temporary file with a blob’s contents

git-var Show a Git logical variable

git-verify-pack Validate packed Git archive files

git-daemon A really simple server for Git repositories

git-fetch-pack Receive missing objects from another repository

git-http-backend Server side implementation of Git over HTTP

git-send-pack Push objects over Git protocol to another repository

git-update-server-info Update auxiliary info file to help dumb servers

git-http-fetch Download from a remote Git repository via HTTP

git-http-push Push objects over HTTP/DAV to another repository

git-receive-pack Receive what is pushed into the repository

git-shell Restricted login shell for Git-only SSH access

git-upload-archive Send archive back to git-archive

git-upload-pack Send objects packed back to git-fetch-pack These are internal helper commands used by other commands; end users typically do not use them directly.

git-check-attr Display gitattributes information

git-check-ignore Debug gitignore / exclude files

git-check-mailmap Show canonical names and email addresses of contacts

git-check-ref-format Ensures that a reference name is well formed

git-column Display data in columns

git-credential Retrieve and store user credentials

git-credential-cache Helper to temporarily store passwords in memory

git-credential-store Helper to store credentials on disk

git-fmt-merge-msg Produce a merge commit message

git-hook Run git hooks

git-interpret-trailers Add or parse structured information in commit messages

git-mailinfo Extracts patch and authorship from a single e-mail message

git-mailsplit Simple UNIX mbox splitter program

git-merge-one-file The standard helper program to use with git-merge-index

git-patch-id Compute unique ID for a patch

git-sh-i18n Git’s i18n setup code for shell scripts

git-sh-setup Common Git shell script setup code

git-stripspace Remove unnecessary whitespace

git core-tutorial A Git core tutorial for developers

git credentials Providing usernames and passwords to Git

git cvs-migration Git for CVS users

git diffcore Tweaking diff output

git everyday A useful minimum set of commands for Everyday Git

git faq Frequently asked questions about using Git

git glossary A Git Glossary

git namespaces Git namespaces

git remote-helpers Helper programs to interact with remote repositories

git submodules Mounting one repository inside another

git tutorial A tutorial introduction to Git

git tutorial-2 A tutorial introduction to Git: part two

git workflows An overview of recommended workflows with Git

git attributes Defining attributes per path

git cli Git command-line interface and conventions

git hooks Hooks used by Git

git ignore Specifies intentionally untracked files to ignore

git mailmap Map author/committer names and/or E-Mail addresses

git modules Defining submodule properties

git format-bundle The bundle file format

git format-chunk Chunk-based file formats

git format-commit-graph Git commit-graph format

git format-index Git index format

git format-pack Git pack format

git format-signature Git cryptographic signature formats

git protocol-capabilities Protocol v0 and v1 capabilities

git protocol-common Things common to various protocols

git protocol-http Git HTTP-based protocols

git protocol-pack How packs are transferred over-the-wire

git protocol-v2 Git Wire Protocol, Version 2

git repository-layout Git Repository Layout

git revisions Specifying revisions and ranges for Git

cd Shows and changes the current directory to a new location

ren Renames files or directories

assoc: fix file associations Associates an extension to open a program or application

mkdir Creates subdirectories within the directories

powercgf Helps detect and troubleshoot battery or power efficiency problems with the device

shutdown Shut down or restart your computer

systeminfo: system information View hardware and software-related information of your Windows device

chkdsk Scans local file system and metadata to find any errors

clip Instantly copies the output of a command

color Changes background colour of Command Prompt window

compact Compress files and folders

copy Copies files from one location to another

del Deletes one or multiple files on your computer instantly

erase Gets rid of specified files

doskey Shows previously used command prompts

driverquery Pulls up a list of drivers installed on the device

exit Exits and closes the current Command Prompt session

pwd Lists the path to the working directory

ls List directory contents

ls -a List contents including hidden files (Files that begin with a dot)

ls -l List contents with more info including permissions (long listing)

ls -r List contents reverse order

touch [filename] Create file

rm [filename] Remove file

rm -i [filename] Remove directory, but ask before

rm -r [dirname] Remove directory

rm -rf [dirname] Remove directory with contents

rm ./\* Remove everything in the current folder

cp [filename] [dirname] Copy file

mv [filename] [dirname] Move file

mv [dirname] [dirname] Move directory

mv [filename] [filename] Rename file or folder

mv [filename] [filename] -v Rename Verbose - print source/destination directory

ipconfig Displays the network details

netstat Provides details of active TCP connections

nslookup Get list of DNS records for a specific domain

arp Change, delete and show Address Resolution Protocol (ARP) information of the devices

nbtstat Displays all the current protocol statistics and current TCP/IP connections

net Find network details and update them

whois Look up domain details of any website

route Check and make alterations to the route table of your Windows machine

tar czvf [dirname].tar.gz [dirname] Create tarball

tar tzvf [dirname] See what is in the tarball

tar xzvf [dirname].tar.gz Extract tarball

history display the history of commands that you have run

docker attach Attach local standard input, output, and error streams to a running container

docker build Build an image from a Dockerfile

docker builder Manage builds

docker checkpoint Manage checkpoints

docker commit Create a new image from a container’s changes

docker config Manage Docker configs

docker container Manage containers

docker context Manage contexts

docker cp Copy files/folders between a container and the local filesystem

docker create Create a new container

docker diff Inspect changes to files or directories on a container’s filesystem

docker events Get real time events from the server

docker exec Run a command in a running container

docker export Export a container’s filesystem as a tar archive

docker history Show the history of an image

docker image Manage images

docker images List images

docker import Import the contents from a tarball to create a filesystem image

docker info Display system-wide information

docker inspect Return low-level information on Docker objects

docker kill Kill one or more running containers

docker load Load an image from a tar archive or STDIN

docker login Log in to a Docker registry

docker logout Log out from a Docker registry

docker logs Fetch the logs of a container

docker manifest Manage Docker image manifests and manifest lists

docker network Manage networks

docker node Manage Swarm nodes

docker pause Pause all processes within one or more containers

docker plugin Manage plugins

docker port List port mappings or a specific mapping for the container

docker ps List containers

docker pull Pull an image or a repository from a registry

docker push Push an image or a repository to a registry

docker rename Rename a container

docker restart Restart one or more containers

docker rm Remove one or more containers

docker rmi Remove one or more images

docker run Run a command in a new container

docker save Save one or more images to a tar archive (streamed to STDOUT by default)

docker search Search the Docker Hub for images

docker secret Manage Docker secrets

docker service Manage services

docker stack Manage Docker stacks

docker start Start one or more stopped containers

docker stats Display a live stream of container(s) resource usage statistics

docker stop Stop one or more running containers

docker swarm Manage Swarm

docker system Manage Docker

docker tag Create a tag TARGET\_IMAGE that refers to SOURCE\_IMAGE

docker top Display the running processes of a container

docker trust Manage trust on Docker images

docker unpause Unpause all processes within one or more containers

docker update Update configuration of one or more containers

docker version Show the Docker version information

docker volume Manage volumes

docker wait Block until one or more containers stop, then print their exit codes

Conda clean

Conda clean -a/ --all Remove index cache, lock files, unused cache packages, tarballs, and logfiles.

Conda clean -i/ --index-cache Remove index cache.

Conda clean -p/--packages Remove unused packages from writable package caches. WARNING: This does not check for packages installed using symlinks back to the package cache.

Conda clean -t/ --tarballs Remove cached package tarballs.

Conda clean -f/--force-pkgs-dirs RemoveÂ allÂ writable package caches.

Conda clean -c/ --tempfiles Remove temporary files that could not be deleted earlier due to being in-use.

Conda clean -l/ --logfiles Remove log files.

Conda clean --json Report all output as json. Suitable for using conda programmatically.

Conda clean -v/ --verbose Can be used multiple times. Once for detailed output, twice for INFO logging, thrice for DEBUG logging, four times for TRACE logging.

Conda clean -q/--quiet Do not display progress bar.

Conda clean -d/--dry-run Only display what would have been done.

Conda clean -y/ --yes Sets any confirmation values to 'yes' automatically. Users will not be asked to confirm any adding, deleting, backups, etc.

Conda compare --file Path to the environment file that is to be compared against.

Conda compare --json Report all output as json. Suitable for using conda programmatically.

Conda compare -v/ --verbose Can be used multiple times. Once for detailed output, twice for INFO logging, thrice for DEBUG logging, four times for TRACE logging.

Conda compare -q/ --quiet Do not display progress bar.

Conda compare -n/ --name Name of environment.

Conda compare -p/ --prefix Full path to environment location (i.e. prefix).

Conda config --system Write to the system .condarc file at '/home/docs/checkouts/readthedocs.org/user\_builds/continuumio-conda/envs/stable/.condarc'.

Conda config --env Write to the active conda environment .condarc file (<no active environment>). If no environment is active, write to the user config file (/home/docs/.condarc).

Conda config --file Write to the given file

Conda config --show Display configuration values as calculated and compiled. If no arguments given, show information for all configuration values.

Conda config --show-sources Display all identified configuration sources.

Conda config --validate Validate all configuration sources. Iterates over all .condarc files and checks for parsing errors.

Conda config --describe Describe given configuration parameters. If no arguments given, show information for all configuration parameters.

Conda config --write-default Write the default configuration to a file. Equivalent toÂ conda config --describe > ~/.condarc.

Conda config --get Get a configuration value.

Conda config --append Add one configuration value to the end of a list key.

Conda config --prepend/ --add Add one configuration value to the beginning of a list key.

Conda config --set Set a boolean or string key.

Conda config --remove "Remove a configuration value from a list key. This removes all instances of the value."

Conda config --remove-key Remove a configuration key (and all its values).

Conda config --stdin Apply configuration information given in yaml format piped through stdin.

Conda create --package\_spec List of packages to install or update in the conda environment.

Conda create --clone Create a new environment as a copy of an existing local environment.

Conda create --file Read package versions from the given file. Repeated file specifications can be passed (e.g. --file=file1 --file=file2).

Conda create --mkdir  is pending deprecation and will be removed in 25.3. Redundant argument.

Conda create --dev UseÂ sys.executable -m condaÂ in wrapper scripts instead of CONDA\_EXE. This is mainly for use during tests where we test new conda sources against old Python versions.

Conda create -n/ --name Name of environment.

Conda create -p/ --prefix Full path to environment location (i.e. prefix).

Conda create -c/ --channel Additional channel to search for packages.

Conda create --use-local Use locally built packages. Identical to '-c local'.

Conda create --override-channels Do not search default or .condarc channels. Requires --channel.

Conda create --repodata-fn Specify file name of repodata on the remote server where your channels are configured or within local backups.

Conda create --experimental "Possible choices: jlap, lock,jlap: Download incremental package index data from repodata.jlap; implies 'lock'. lock: use locking when reading, updating index (repodata.json) cache. Now enabled."

Conda create --no-lock Disable locking when reading, updating index (repodata.json) cache.

Conda create --repodata-use-zst/ --no-repodata-use-zst Check for/do not check for repodata.json.zst. Enabled by default. (default: Null)

Conda create --subdir/ --platform Use packages built for this platform. The new environment will be configured to remember this choice.

Conda create --strict-channel-priority Packages in lower priority channels are not considered if a package with the same name appears in a higher priority channel.

Conda create --no-channel-priority Package version takes precedence over channel priority. Overrides the value given byÂ conda config --show channel\_priority.

Conda create --no-deps Do not install, update, remove, or change dependencies. This WILL lead to broken environments and inconsistent behavior. Use at your own risk.

Conda create --only-deps Only install dependencies.

Conda create --no-pin Ignore pinned file.

Conda create --no-default-packages Ignore create\_default\_packages in the .condarc file.

Conda create --solver "Possible choices: classic, Choose which solver backend to use."

Conda create --copy Install all packages using copies instead of hard- or soft-linking.

Conda create --no-shortcuts Don't install start menu shortcuts

Conda create --shortcuts-only Install shortcuts only for this package name. Can be used several times.

Conda create -C/ --use-index-cache Use cache of channel index files, even if it has expired. This is useful if you don't want conda to check whether a new version of the repodata file exists, which will save bandwidth.

Conda create -k/ --insecure "Allow conda to perform ""insecure"" SSL connections and transfers. Equivalent to setting 'ssl\_verify' to 'false'."

Conda create --offline Offline mode. Don't connect to the Internet.

Conda create -d/ --dry-run Only display what would have been done.

Conda create -y/ --yes Sets any confirmation values to 'yes' automatically. Users will not be asked to confirm any adding, deleting, backups, etc.

Conda create --download-only Solve an environment and ensure package caches are populated, but exit prior to unlinking and linking packages into the prefix.

Conda create --show-channel-urls Show channel urls. Overrides the value given byÂ conda config --show show\_channel\_urls.

Conda doctor -v/ --verbose Can be used multiple times. Once for detailed output, twice for INFO logging, thrice for DEBUG logging, four times for TRACE logging.

Conda doctor -n/ --name Name of environment.

Conda doctor -p/ --prefix Full path to environment location (i.e. prefix).

Conda env command Possible choices: config, create, export, list, remove, update

Conda env config Configure a conda environment.

Conda env config vars Interact with environment variables associated with Conda environments.

Conda env config vars list List environment variables for a conda environment.

Conda env config vars list -n/ --name Name of environment.

Conda env config vars list -p/ --prefix Full path to environment location (i.e. prefix).

Conda env config vars set Set environment variables for a conda environment.

Conda env config vars set -n/ --name Name of environment.

Conda env config vars set -p/ --prefix Full path to environment location (i.e. prefix).

Conda env config vars unset Unset environment variables for a conda environment.

Conda env config vars unset -n/ --name Name of environment.

Conda env config vars unset -p/ --prefix Full path to environment location (i.e. prefix).

Conda env create "Create an environment based on an environment definition file. remote\_definition, Remote environment definition / IPython notebook"

Conda env create -f/ --file Environment definition file (default: environment.yml)

Conda env create --no-default-packages Ignore create\_default\_packages in the .condarc file.

Conda env create --solver "Possible choices: classic, Choose which solver backend to use."

Conda env create --subdir/ --platform Use packages built for this platform. The new environment will be configured to remember this choice. Should be formatted like 'osx-64', 'linux-32', 'win-64', and so on. Defaults to the current (native) platform.

Conda env create -n/ --name Name of environment.

Conda env create -p/ --prefix Full path to environment location (i.e. prefix).

Conda env create -C/ --use-index-cache Use cache of channel index files, even if it has expired. This is useful if you don't want conda to check whether a new version of the repodata file exists, which will save bandwidth.

Conda env create -k/ --insecure "Allow conda to perform ""insecure"" SSL connections and transfers. Equivalent to setting 'ssl\_verify' to 'false'."

Conda env create --offline Offline mode. Don't connect to the Internet.

Conda env create -d/ --dry-run Only display what would have been done.

Conda env create -y/ --yes Sets any confirmation values to 'yes' automatically. Users will not be asked to confirm any adding, deleting, backups, etc.

ASSOC Displays or modifies file extension associations.

ATTRIB Displays or changes file attributes.

BREAK Sets or clears extended CTRL+C checking.

BCDEDIT Sets properties in boot database to control boot loading.

CACLS Displays or modifies access control lists (ACLs) of files.

CALL Calls one batch program from another.

CD Displays the name of or changes the current directory.

CHCP Displays or sets the active code page number.

CHDIR Displays the name of or changes the current directory.

CHKDSK Checks a disk and displays a status report.

CHKNTFS Displays or modifies the checking of disk at boot time.

CLS Clears the screen.

CMD Starts a new instance of the Windows command interpreter.

COLOR Sets the default console foreground and background colors.

COMP Compares the contents of two files or sets of files.

COMPACT Displays or alters the compression of files on NTFS partitions.

CONVERT Converts FAT volumes to NTFS. You cannot convert the

current drive.

COPY Copies one or more files to another location.

DATE Displays or sets the date.

DEL Deletes one or more files.

DIR Displays a list of files and subdirectories in a directory.

DISKPART Displays or configures Disk Partition properties.

DOSKEY Edits command lines, recalls Windows commands, and

creates macros.

DRIVERQUERY Displays current device driver status and properties.

ECHO Displays messages, or turns command echoing on or off.

ENDLOCAL Ends localization of environment changes in a batch file.

ERASE Deletes one or more files.

EXIT Quits the CMD.EXE program (command interpreter).

FC Compares two files or sets of files, and displays the

differences between them.

FIND Searches for a text string in a file or files.

FINDSTR Searches for strings in files.

FOR Runs a specified command for each file in a set of files.

FORMAT Formats a disk for use with Windows.

FSUTIL Displays or configures the file system properties.

FTYPE Displays or modifies file types used in file extension

associations.

GOTO Directs the Windows command interpreter to a labeled line in

a batch program.

GPRESULT Displays Group Policy information for machine or user.

GRAFTABL Enables Windows to display an extended character set in

graphics mode.

HELP Provides Help information for Windows commands.

ICACLS Display, modify, backup, or restore ACLs for files and

directories.

IF Performs conditional processing in batch programs.

LABEL Creates, changes, or deletes the volume label of a disk.

MD Creates a directory.

MKDIR Creates a directory.

MKLINK Creates Symbolic Links and Hard Links

MODE Configures a system device.

MORE Displays output one screen at a time.

MOVE Moves one or more files from one directory to another

directory.

OPENFILES Displays files opened by remote users for a file share.

PATH Displays or sets a search path for executable files.

PAUSE Suspends processing of a batch file and displays a message.

POPD Restores the previous value of the current directory saved by

PUSHD.

PRINT Prints a text file.

PROMPT Changes the Windows command prompt.

PUSHD Saves the current directory then changes it.

RD Removes a directory.

RECOVER Recovers readable information from a bad or defective disk.

REM Records comments (remarks) in batch files or CONFIG.SYS.

REN Renames a file or files.

RENAME Renames a file or files.

REPLACE Replaces files.

RMDIR Removes a directory.

ROBOCOPY Advanced utility to copy files and directory trees

SET Displays, sets, or removes Windows environment variables.

SETLOCAL Begins localization of environment changes in a batch file.

SC Displays or configures services (background processes).

SCHTASKS Schedules commands and programs to run on a computer.

SHIFT Shifts the position of replaceable parameters in batch files.

SHUTDOWN Allows proper local or remote shutdown of machine.

SORT Sorts input.

START Starts a separate window to run a specified program or command.

SUBST Associates a path with a drive letter.

SYSTEMINFO Displays machine specific properties and configuration.

TASKLIST Displays all currently running tasks including services.

TASKKILL Kill or stop a running process or application.

TIME Displays or sets the system time.

TITLE Sets the window title for a CMD.EXE session.

TREE Graphically displays the directory structure of a drive or

path.

TYPE Displays the contents of a text file.

VER Displays the Windows version.

VERIFY Tells Windows whether to verify that your files are written

correctly to a disk.

VOL Displays a disk volume label and serial number.

XCOPY Copies files and directory trees.

WMIC Displays WMI information inside interactive command shell.

ifconfig - Display network interfaces and IP addresses

traceroute - Trace all the network hops to reach the destination

wget - Direct download files from the internet

ufw - Firewall command

iptables - Base firewall for all other firewall utilities to interface with

apt, pacman, yum, rpm - Package managers depending on the distribution

sudo - Command to escalate privileges in Linux

cal - View a command-line calendar

alias - Create custom shortcuts for your regularly used commands

dd - Majorly used for creating bootable USB sticks

whereis - Locate the binary, source, and manual pages for a command

whatis - Find what a command is used for

top - View active processes live with their system usage

useradd and usermod - Add a new user or change existing user data

passwd - Create or update passwords for existing users

zip - Zip files in Linux

unzip - Unzip files in Linux

ssh - Secure Shell command in Linux

service - Linux command to start and stop services

ps - Display active processes

kill and killall - Kill active processes by process ID or name

df - Display disk filesystem information

mount - Mount file systems in Linux

chmod - Command to change file permissions

chown - Command for granting ownership of files or folders

echo - Print any text that follows the command

less - Linux command to display paged outputs in the terminal

man - Access manual pages for all Linux commands

uname - Linux command to get basic information about the OS

whoami - Get the active username

tar - Command to extract and compress files in linux

grep - Search for a string within an output

head - Return the specified number of lines from the top

tail - Return the specified number of lines from the bottom

diff - Find the difference between two files

cmp - Allows you to check if two files are identical

comm - Combines the functionality of diff and cmp

sort - Linux command to sort the content of a file while outputting

export - Export environment variables in Linux

ls - The most frequently used command in Linux to list directories

pwd - Print working directory command in Linux

cd - Linux command to navigate through directories

mkdir - Command used to create directories in Linux

mv - Move or rename files in Linux

cp - Similar usage as mv but for copying files in Linux

rm - Delete files or directories

touch - Create blank/empty files

ln - Create symbolic links (shortcuts) to other files

clear - Clear the terminal display

cat - Display file contents on the terminal