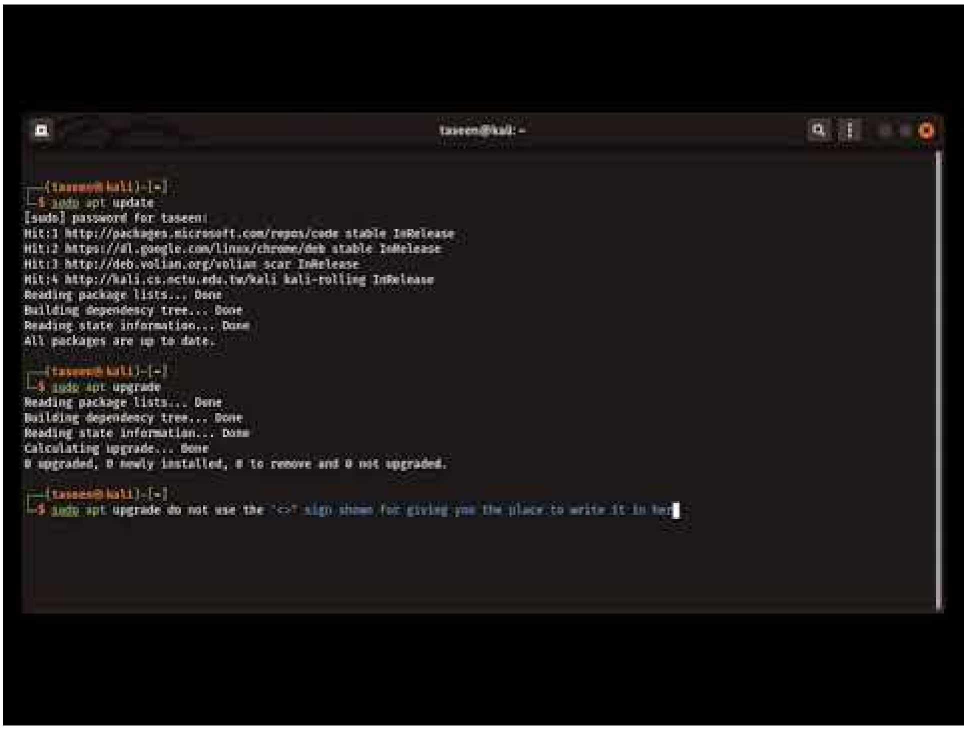
MD Support PDF

APT sucks, use Nala instead!

mdsupportforall.blogspot.com/2022/03/apt-sucks-use-nala-instead.html



```
┌──(taseen㉿kali)-[~]
└─$ sudo apt update
[sudo] password for taseen:
Hit:1 http://packages.microsoft.com/repos/code stable InRelease
Hit:2 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:3 http://deb.volian.org/volian scar InRelease
Hit:4 http://kali.cs.nctu.edu.tw/kali kali-rolling InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.

┌──(taseen㉿kali)-[~]
└─$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

┌──(taseen㉿kali)-[~]
└─$ sudo apt upgrade do not use the `<>` sign shown for giving you the place to write it in her
```

APT Documentation

What is APT ??

apt is a command-line utility for installing, updating, removing, and otherwise managing deb packages on Ubuntu, Debian, and related Linux distributions. It combines the most frequently used commands from the apt-get and apt-cache tools with different default values of some options.

apt is designed for interactive use. Prefer using apt-get and apt-cache in your shell scripts as they are backward compatible between the different versions and have more options and features.

Most of the apt commands must be run as a user with sudo privileges.

This guide serves as a quick reference for the apt commands.

Updating package index (apt update)

The APT package index is basically a database that holds records of available packages from the repositories enabled in your system.

To update the package index run the command below. This will pull the latest changes from the APT repositories:

```
sudo apt update
```

Always update the package index before upgrading or installing new packages.

Upgrading packages (apt upgrade)

Regularly updating your Linux system is one of the most important aspects of overall system security.

To upgrade the installed packages to their latest versions run:

```
sudo apt upgrade
```

The command doesn’t upgrade packages that require removal of installed packages.

If you want to upgrade a single package, pass the package name:

```
sudo apt upgrade package_name
```

It is always a good idea to configure automatic security updates .

Full Upgrading (apt full-upgrade)

The difference between upgrade and full-upgrade is that the later will remove the installed packages if that is needed to upgrade the whole system.

```
sudo apt full-upgrade
```

Be extra careful when using this command.

Installing packages (apt install)

Installing packages is as simple as running the following command:

```
sudo apt install package_name
```

If you want to install multiple packages with one command, specify them as a space-separated list:

```
sudo apt install package1 package2
```

To install local deb files provide the full path to file. Otherwise, the command will try to retrieve and install the package from the APT repositories.

```
sudo apt install /full/path/file.deb
```

Removing Packages (apt remove)

To remove an installed package type the following:

```
sudo apt remove package_name
```

You can also specify multiple packages, separated by spaces:

```
sudo apt remove package1 package2
```

The remove command will uninstall the given packages, but it may leave some configuration files behind. If you want to remove the package including all configuration files, use purge instead of remove :

```
sudo apt purge package_name
```

Remove Unused Packages (apt autoremove)

Whenever a new package that depends on other packages is installed on the system, the package dependencies will be installed too. When the package is removed, the dependencies will stay on the system. This leftover packages are no longer used by anything else and can be removed.

To remove the unneeded dependencies use the following command:

```
sudo apt autoremove
```

Listing Packages (apt list)

The list command allows you to list the available, installed and, upgradeable packages.

To list all available packages use the following command:

```
sudo apt list
```

The command will print a list of all packages, including information about the versions and architecture of the package. To find out whether a specific package is installed, you can filter the output with the grep command.

```
sudo apt list | grep package_name
```

To list only the installed packages type:

```
sudo apt list --installed
```

Getting a list of the upgradeable packages may be useful before actually upgrading the packages:

```
sudo apt list --upgradeable
```

Searching Packages (apt search)

This command allows you to search for a given package in the list of the available packages:

```
sudo apt search package_name
```

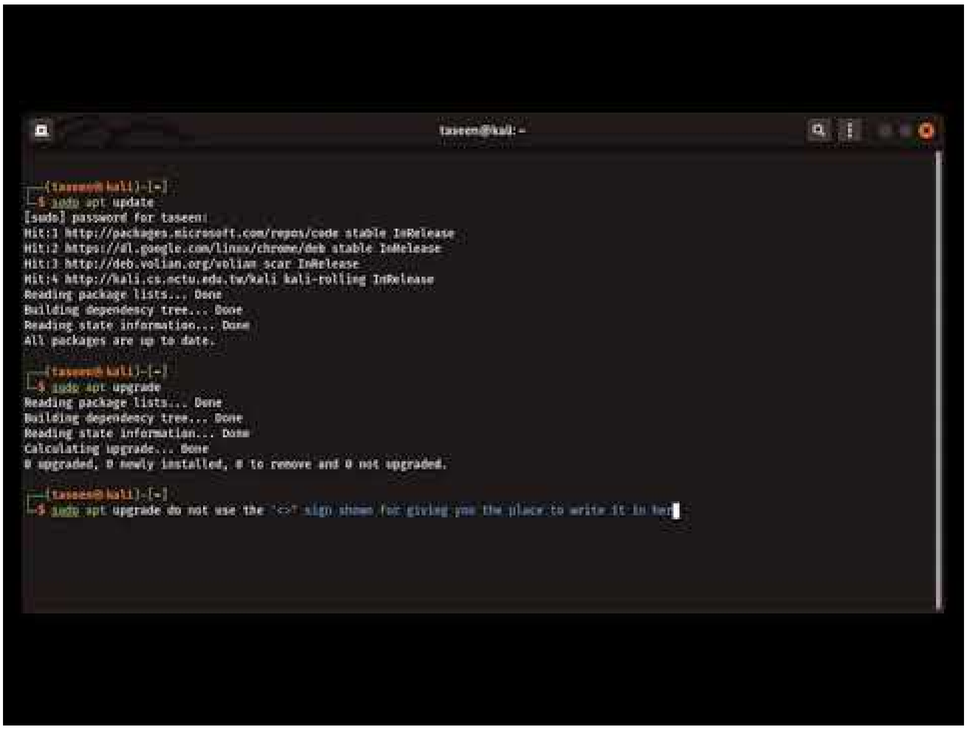
If found, the command will return the packages which name matches the search term.

Package Information (apt show)

The information about the package dependencies, installation size, the package source, and so on might be useful before removing or installing a new package.

To retrieve information about a given package, use the show command:

```
sudo apt show package_name
```



```
┌──(taseen㉿kali)-[~]
└─$ sudo apt update
[sudo] password for taseen:
Hit:1 http://packages.microsoft.com/repos/code stable InRelease
Hit:2 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:3 http://deb.volian.org/volian scar InRelease
Hit:4 http://kali.cs.nctu.edu.tw/kali kali-rolling InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.

┌──(taseen㉿kali)-[~]
└─$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

┌──(taseen㉿kali)-[~]
└─$ sudo apt upgrade do not use the '<>' sign shown for giving you the place to write it in her
```

Watch Video At: https://youtu.be/j45zgix9fpM

Nala Documentation

What is Nala ??

Nala is a **front-end** for libapt-pkg. Specifically we interface using the python-apt api.

Especially for newer users it can be hard to understand what apt is trying to do when installing or upgrading.

We aim to solve this by not showing some redundant messages, formatting the packages better, and using color to

show specifically what will happen with a package during **install**, **removal**, or an **upgrade**.

usage: nala [--options] <command>

commands:

install	install packages
remove	remove packages
purge	purge packages

update update package list and upgrade the system

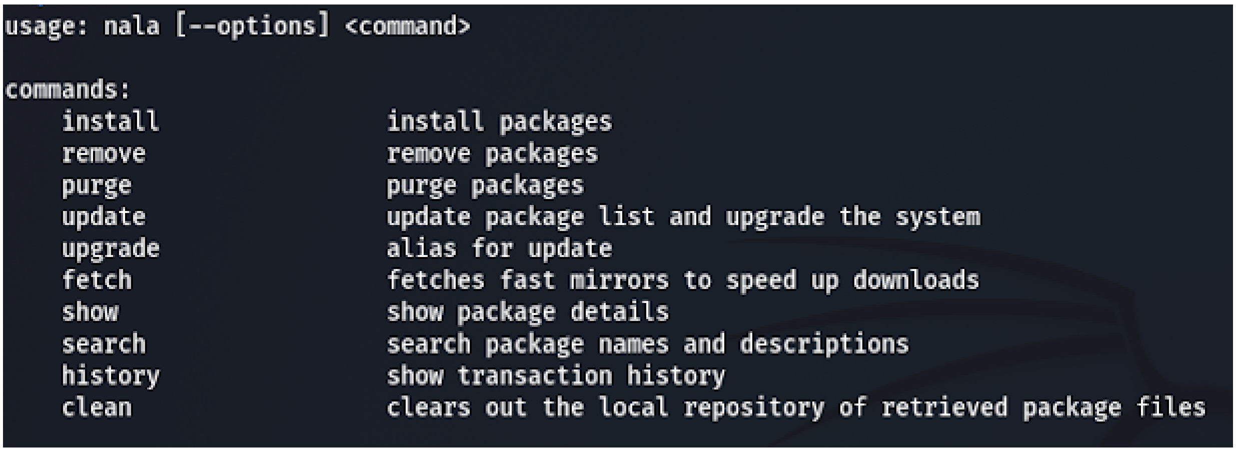
upgrade alias for update

fetch fetches fast mirrors to speed up downloads

show show package details

history show transaction history

clean clears out the local repository of retrieved package files



```
usage: nala [--options] <command>

commands:
    install                 install packages
    remove                  remove packages
    purge                   purge packages
    update                  update package list and upgrade the system
    upgrade                 alias for update
    fetch                   fetches fast mirrors to speed up downloads
    show                    show package details
    search                  search package names and descriptions
    history                 show transaction history
    clean                   clears out the local repository of retrieved package files
```

optional arguments:

-h, --help show this help message and exit

-y, --assume-yes assume 'yes' to all prompts and run non-interactively

-d, --download-only package files are only retrieved, not unpacked or installed

-v, --verbose logs extra information for debugging

--no-update skips updating the package list

--no-autoremove stops nala from autoremoving packages

--remove-essential allows the removal of essential packages

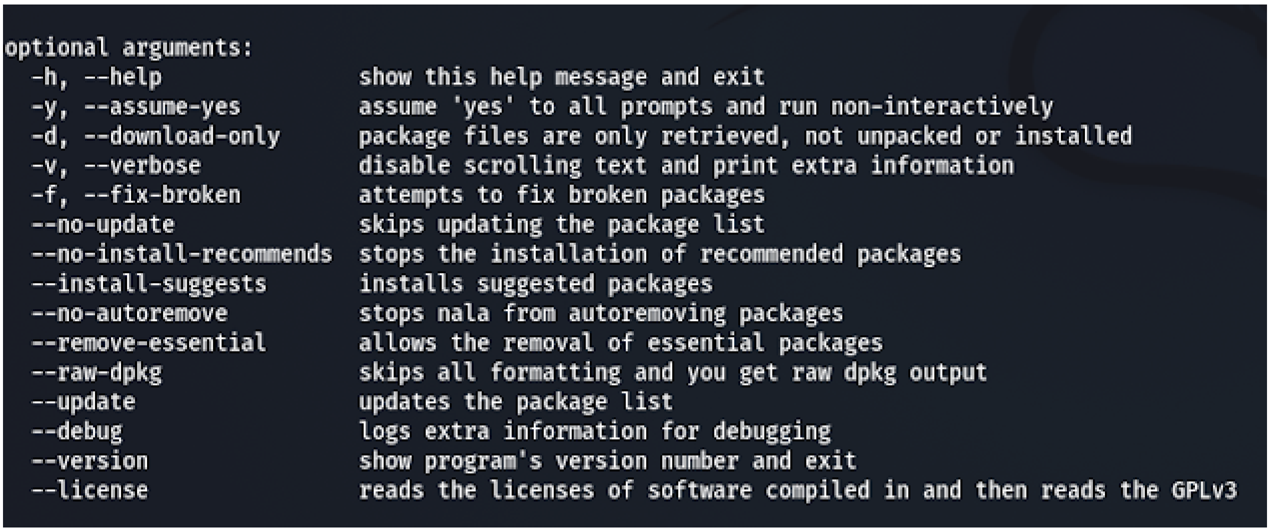
--raw-dpkg skips all formatting and you get raw dpkg output

--update updates the package list

--debug logs extra information for debugging

--version show program's version number and exit

--license	reads the licenses of software compiled in and then reads the GPLv3



```
optional arguments:
  -h, --help              show this help message and exit
  -y, --assume-yes        assume 'yes' to all prompts and run non-interactively
  -d, --download-only     package files are only retrieved, not unpacked or installed
  -v, --verbose           disable scrolling text and print extra information
  -f, --fix-broken        attempts to fix broken packages
  --no-update             skips updating the package list
  --no-install-recommends stops the installation of recommended packages
  --install-suggests      installs suggested packages
  --no-autoremove         stops nala from autoremoving packages
  --remove-essential      allows the removal of essential packages
  --raw-dpkg              skips all formatting and you get raw dpkg output
  --update                updates the package list
  --debug                 logs extra information for debugging
  --version               show program's version number and exit
  --license               reads the licenses of software compiled in and then reads the GPLv3
```

Steps to install:

> git clone https://gitlab.com/volian/nala

> cd nala

> python3 ./setup.py build

> sudo python3 ./setup.py install

It also has a apt repo:

> echo "deb [arch=amd64,arm64,armhf] http://deb.volian.org/volian/ scar main" | sudo tee /etc/apt/sources.list.d/volian-archive-scar-unstable.list

> wget -qO - https://deb.volian.org/volian/scar.key | sudo tee /etc/apt/trusted.gpg.d/volian-archive-scar-unstable.gpg > /dev/null
>
> sudo apt update && sudo apt install nala