

# How to install and remove software

---

What is the difference between APT and DNF?

APT and DNF are package managers for different distro families. APT is Debian/Ubuntu's using .deb packages, while DNF is Fedora/RHEL/CentOS and uses .rpm packages.

## Installing software

How do you find software?

```
dnf search package  
  
package-name.x86-64          Package that does...
```

In both **dnf** and **apt**, you can use the **search** feature in order to find specific software or packages.

How do you install software?

```
sudo dnf install package-name  
[sudo] password for user:  
(installation process)  
Complete!
```

Using the name found in **search**, **install** to download and install the specified package with dependencies. This requires **sudo** privileges.

How do you remove software?

### Removal

```
sudo dnf remove package-name
```

To remove a package, use **remove**, though config files

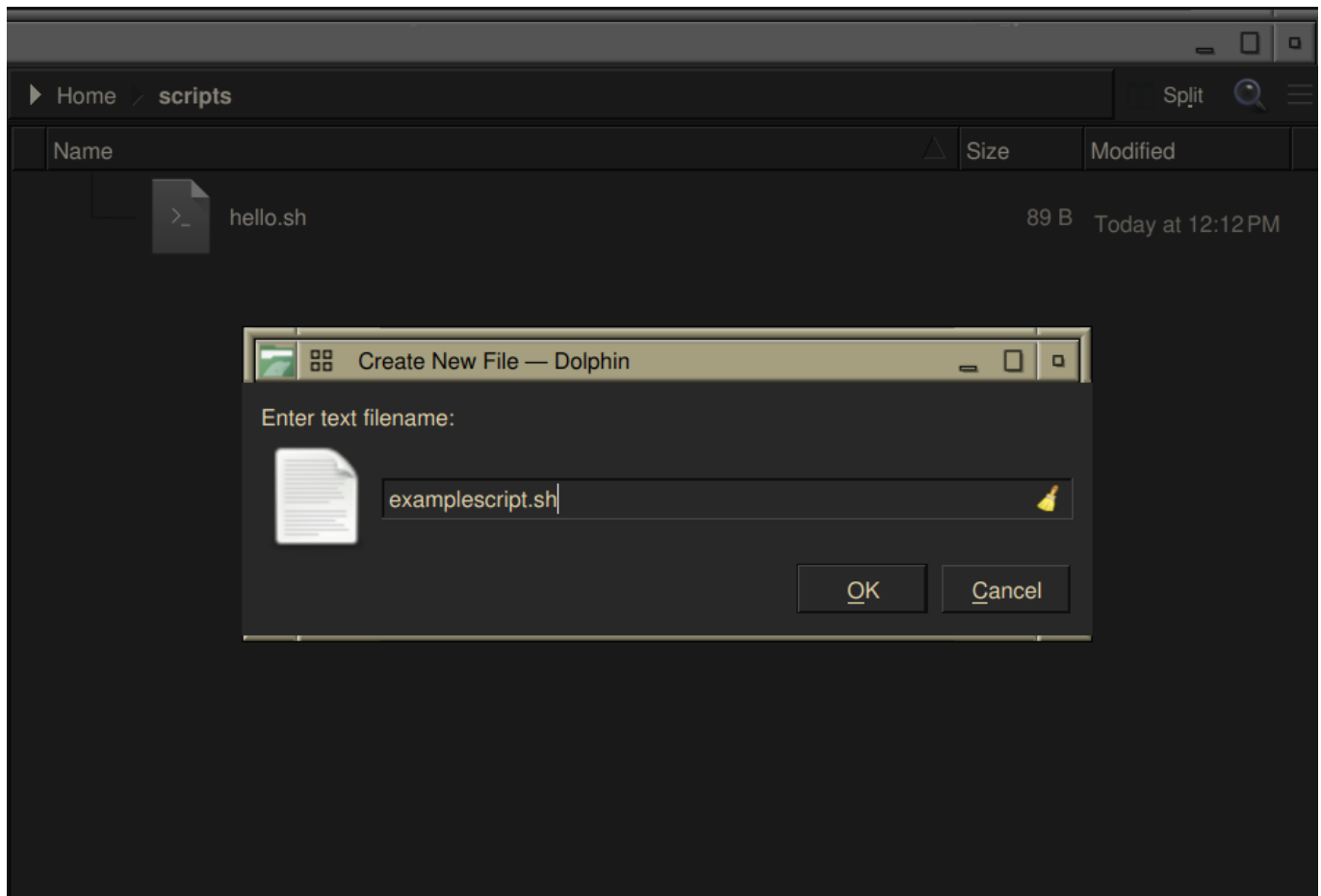
### Cache Cleaning

```
sudo dnf clean all && sudo dnf autoremove
```

## Shell Scripting

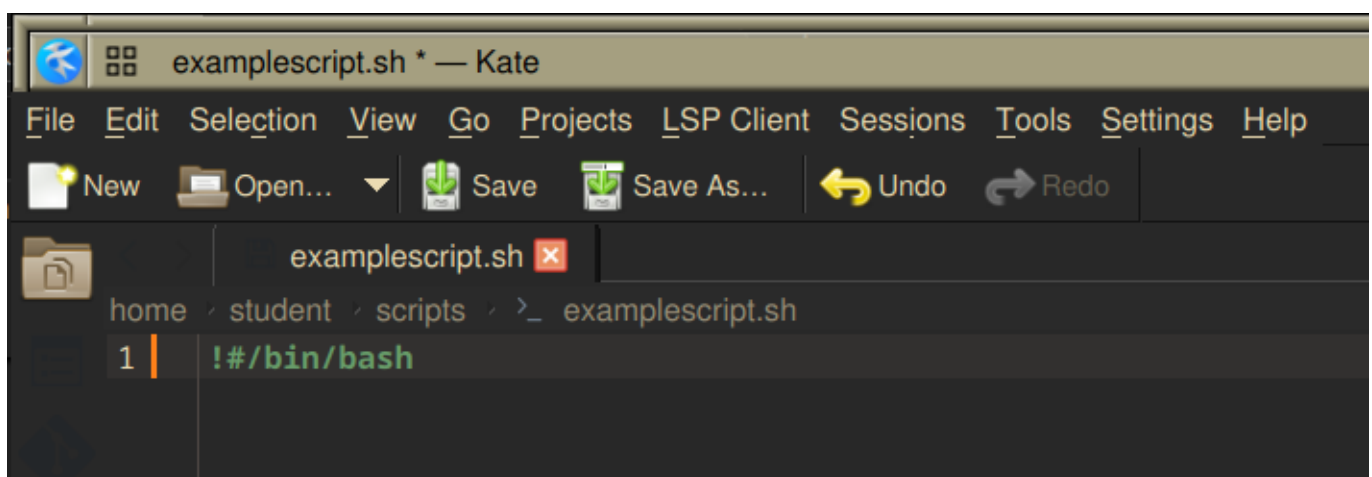
## Step 1: Create the Script File

Create a file with the `.sh` extension:



## Step 2: Add the Shebang

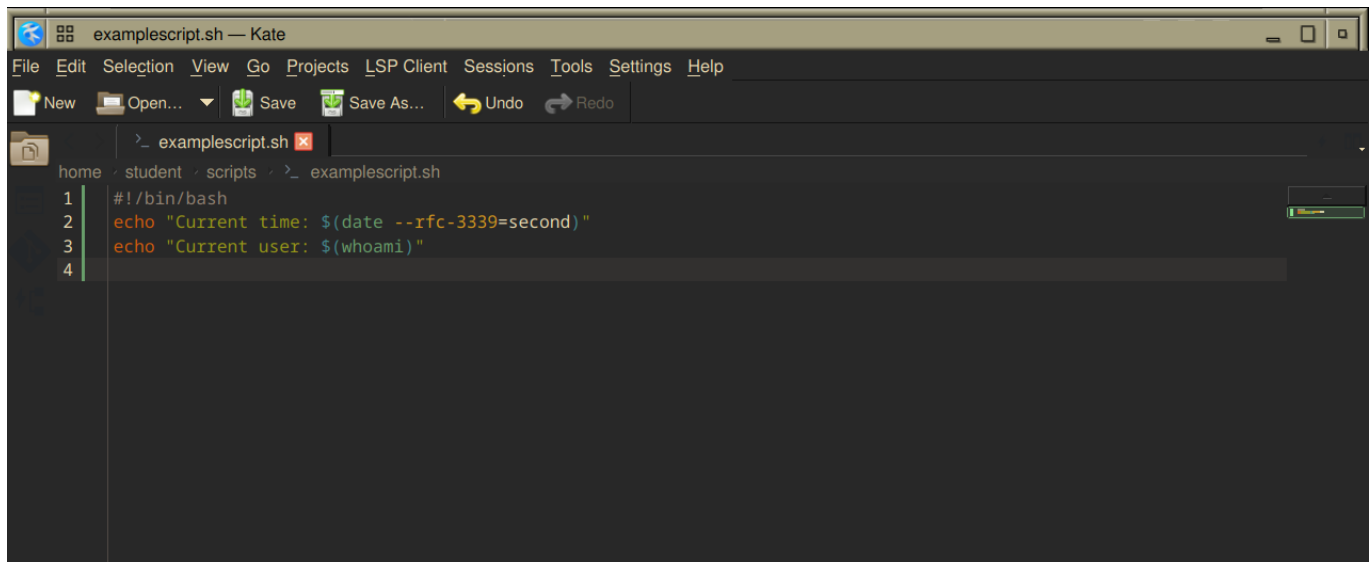
Open the file in a text editor and add the shebang line as the first line:



This tells the system to execute the script using the Bash interpreter located at `/bin/bash`.

## Step 3: Add Commands

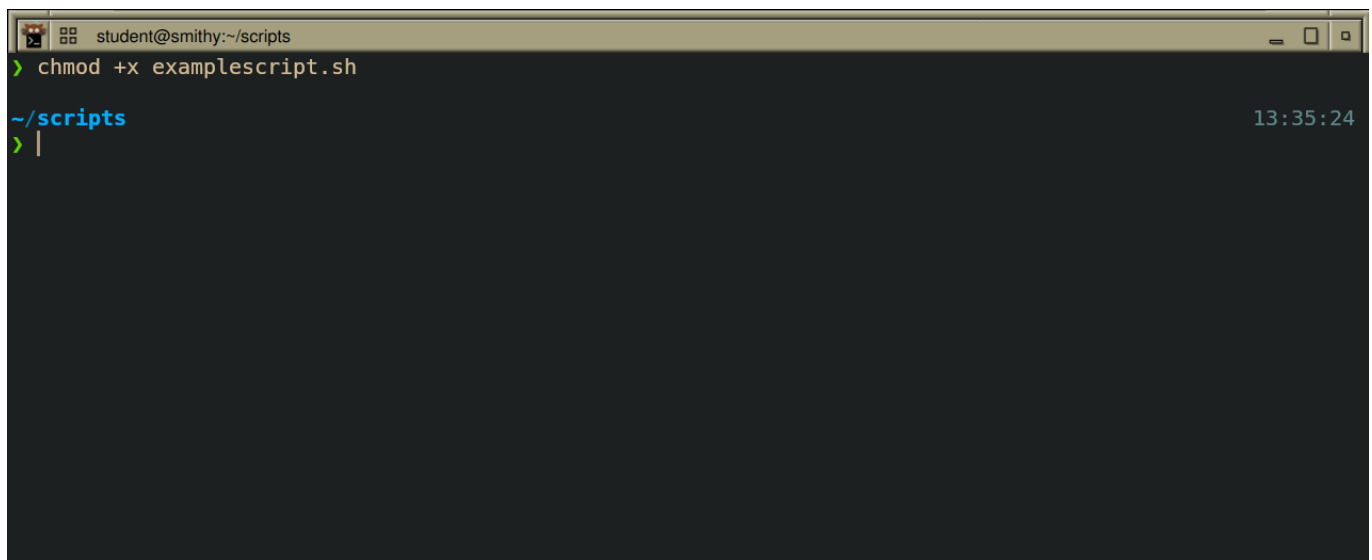
Write your commands line by line, just as you would in the terminal:

A screenshot of a text editor window titled 'examplescript.sh — Kate'. The window has a menu bar with 'File', 'Edit', 'Selection', 'View', 'Go', 'Projects', 'LSP Client', 'Sessions', 'Tools', 'Settings', and 'Help'. Below the menu bar is a toolbar with icons for 'New', 'Open...', 'Save', 'Save As...', 'Undo', and 'Redo'. The main editing area shows a file named 'examplescript.sh' with the following content:

```
home / student / scripts / >_ examplescript.sh
1  #!/bin/bash
2  echo "Current time: $(date --rfc-3339=second)"
3  echo "Current user: $(whoami)"
4
```

## Step 4: Make it Executable

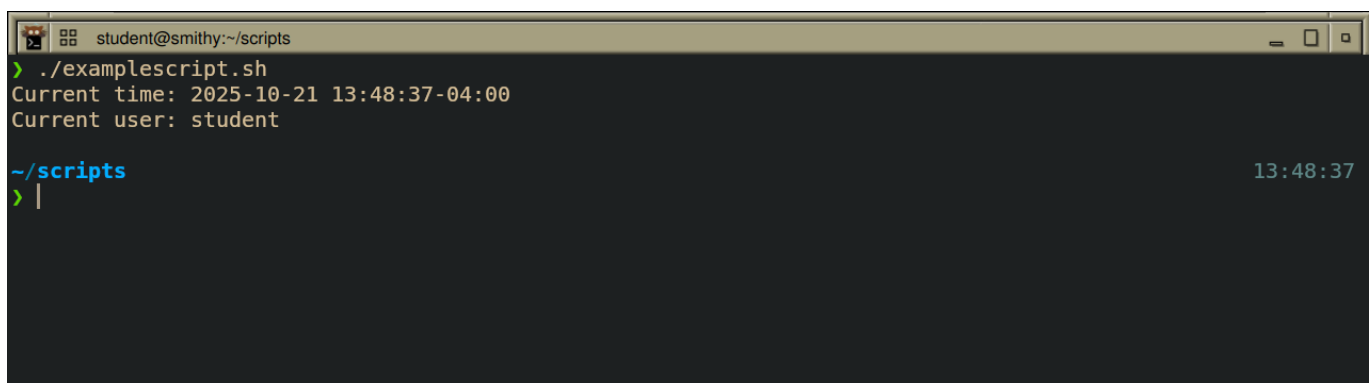
Before running the script, give it execute permissions:

A screenshot of a terminal window with the title 'student@smithy:~/scripts'. The prompt is '>'. The user has entered the command 'chmod +x examplescript.sh'. The terminal shows the command was executed successfully. The prompt is now '~-/scripts' and the time '13:35:24' is displayed in the top right corner.

```
student@smithy:~/scripts
> chmod +x examplescript.sh
~/scripts 13:35:24
> |
```

## Step 5: Run the script

Execute the script using either **bash** or **./**:

A screenshot of a terminal window with the title 'student@smithy:~/scripts'. The prompt is '>'. The user has entered the command './examplescript.sh'. The terminal shows the output of the script: 'Current time: 2025-10-21 13:48:37-04:00' and 'Current user: student'. The prompt is now '~-/scripts' and the time '13:48:37' is displayed in the top right corner.

```
student@smithy:~/scripts
> ./examplescript.sh
Current time: 2025-10-21 13:48:37-04:00
Current user: student
~/scripts 13:48:37
> |
```