# [What Is the Bash Shell, and Why Is It So Important to Linux?](https://www.howtogeek.com/726559/what-is-the-bash-shell-and-why-is-it-so-important-to-linux/)

1. When you open a terminal window and type commands, something has to take what you’ve typed, figure out what you intended, and run the tasks you asked for.
2. The software that does this is the shell.
3. A shell is a command interpreter.
4. It scans what you’ve typed and picks out the commands, directory names, file names, and program names so that it can figure out what you’re trying to achieve.

People often use the phrases “terminal windows,” “command line,” and “shell” interchangeably, but they’re three distinct things. A terminal window is a software representation of a physical [teletype terminal](https://www.howtogeek.com/428174/what-is-a-tty-on-linux-and-how-to-use-the-tty-command/).

It gives you a connection to the computer. In order to do anything useful, you must be able to type instructions at a command line.

The command line is provided by the shell, and the terminal window lets you access the shell.

Shells also allow you to parcel up a collection of commands into a text file called a script.

All the commands in the script are executed for you each time you run the script. Scripts deliver efficiency, repeatability, and convenience.

The first [Unix](https://www.howtogeek.com/182649/htg-explains-what-is-unix/) shell was the [Thompson shell](https://en.wikipedia.org/wiki/Thompson_shell), called sh.

It was written by [Ken Thompson](https://en.wikipedia.org/wiki/Ken_Thompson), who is possibly the most key member of the original Unix founding fathers at [Bell Labs](https://en.wikipedia.org/wiki/Bell_Labs).

The Thompson shell was used as the default Unix shell up to and including Unix Version 6. It was replaced by the [Bourne shell](https://en.wikipedia.org/wiki/Bourne_shell) in Version 7 of Unix in 1979.

## 

## What is Scripting

Now, imagine that the execution of all the above commands is your daily task. Every day you are required to execute all of the above commands without fail as well as store the observed information. Soon enough this will become an extremely tedious task destined for failure.

Thus the obvious notion is to think of some way to execute all given commands together. This is where scripting becomes your salvation.

To see what is meant by scripting, use shell in combination with your favorite text editor eg. [vi](https://linuxconfig.org/vim-tutorial) to create a new file called **task.sh** containing all the above commands, each on a separate line.

Once ready, make your new file executable using **chmod** command with an option **+x**. Lastly, execute your new script by prefixing its name with **./**.

As you can see, by use of scripting, any shell interaction can be automated and scripted. Furthermore, it is now possible to automatically execute our new shell script **task.sh** daily at any given time by use of [cron time-based job scheduler](https://linuxconfig.org/linux-cron-guide) and store the script’s output to a file every time it is executed.

However, this is a tale for an another day, for now let’s just concentrate on a task ahead.

What is Bash

So far we have covered shell and scripting. What about Bash? Where does the bash fit in? As already mentioned, the bash is a default interpreter on many GNU/Linux systems, thus we have been using it even without realising. This is why our previous shell script works even without us defining bash as an interpreter. To see what is your default interpreter execute command **echo $SHELL**:

$ echo $SHELL

/bin/bash

There are various other shell interpreters available, such as Korn shell, C shell and more. From this reason, it is a good practice to define the shell interpreter to be used explicitly to interpret the script’s content.

To define your script’s interpreter as Bash, first locate a full path to its executable binary using **which** command, prefix it with a [shebang](https://en.wikipedia.org/wiki/Shebang_%28Unix%29) **#!** and insert it as the first line of your script. There are various other techniques how to define shell interpreter, but this is a solid start.