

1 Introduction

Terminals! What is the terminal? What we call "the terminal" now days, is actually a terminal emulator software. A software that emulates the functionality of a terminal. And an actual terminal is a physical device that is attached to a computer and used for sending and receiving data back and forth.

2 Old terminals

Back in the day, when there were computers large as a room, to control such computers in the mid-1970s they used a device called the terminal, like I said. A terminal looks like a small computer with a keyboard attached to it although the terminal is not a computer, it has barely enough functionality to handle keyboard input. The terminal depends on the host (computer that it's connected to) for processing power.

By the way a terminal that has some local programmable data-processing capability is called a "smart terminal" or a fat client. A terminal that fully depends on the host's computing power can be called a "dumb terminal" or a thin client.

3 Terminal emulators

Today we don't have this large computers anymore. We have personal computers and laptops which have software called the terminal emulators or "terminals" as we call them. They are programs used to gain deeper control over your operating system. If you can do a lot with a user interface or a GUI, you can do thousand times more with a terminal. It has more capability and control. And for us as programmers we can use it to run terminal commands. For example: we can use the terminal to run a Python script, or to install NPM packages, move files around, and what not.

Much like the old terminals, today our terminals also depend on the host. Terminal emulator only controls user-input and display of characters on the screen and what runs all of our commands is called the shell.

4 The shell

Shell is a software that basically connects the kernel and the user. What is a kernel? Think of it like brains of an operating system, the most important component in your OS. One in charge of running all of the tasks and processes on your machine. A shell can be used to pass commands to the kernel for it to execute it and return output. We can access the shell using the terminal. When the shell software starts it shows us a string of text which can often look like a \$ symbol or a > and a blinking cursor next to it, that is called a prompt! A prompt is basically exactly what you think, when you call an input function in Python or prompt function in JavaScript, the functionality of both is almost exactly the same. Once you input/type your command into the shell, the shell then parses your command (basically cuts it up) and passes it to the kernel. From there the kernel takes over the rest.

5 Summary

So yeah... That's basically how terminals, shells and the kernel work together in modern and old times.