**Text Editors and IDEs**

Choosing the right text editor can be tricky sometimes. Therefore, you have to consider some features that the text editor can provide to pick one. First, it should have a code autocompletion feature, which will save you time. Second, text highlighting, this feature colors the code with different colors according to the code nature and type—next, the ability to change the editor's general theme. Transforming the editing window and its various components' color will positively impact the developer's visual health since it is considered an eye comfort mechanism. Finally, to be supported with a wide range of extensions that make life way more manageable and the coder performance more productive and fruitful.

There are many text editors out there, and you can choose to work with any you feel most comfortable. Usually, any operating system comes with their text editor installed; however, these editors lack the productive tools and the hyper-functional capabilities third party editors have since they do not have professional features. Instead, one could use one of the third-party editors. Some of them are free, and others you can purchase online. Generally, most of them are exceptionally well designed and come with all great functionalities any programmer would want to have, yet you have to try some, then decide which one suits you the most and stick to it, because it entirely relies on personal preference.

Text editors are not IDEs. IDE stands for Integrated Development Environment; it can have more than a text editor. IDEs have compilers specific to the language your code is typed in, a debugger to fix bugs and spot errors in your code, a filing system that allows you to manage and navigate between your app files and folders, and a linker, which is a piece of software that will enable you to link different object files so that the application will work properly.

**Command-line interface**

Command-line interface, or CLI for short, is a text-based interface where you can write your desired input command and probably get back outputs that tell you about your running command status. You can also have multiple command lines opened and working concurrently; thus, you can perform different tasks simultaneously.

Opening the terminal depends solely on your operating system. Usually, on macOS and Linux, it's called Terminal. On windows, it is better to log in to other third-party terminals.

The shell is the part of the operating system that tells the terminal how to behave and perform tasks delegated to the CLI. The most common shell used in the tech industry is Bash, and it stands for Bourne again shell.

Navigation through directories and files is one of the essential functionalities the terminal allows you to do. Many tasks highly depend on the ability to locate and referencing locations in the memory. Some of the main commands to navigate the system are:

* pwd: Print working directory. This command does not take any argument.
* ls: lists given directory files and folders. Can take arguments and options
* cd: change the current directory.

These commands are the very basics; however, they can receive arguments and options to specify how the commands should be executed and on which directory. We can identify any desired location in two main ways, whether according to their relative or absolute paths. Before we learn what they are, we should learn some basics about our file system initially. Linux file system organizes machine directories in a tree structure that form a sort of hierarchy. The topmost element in the structure is called the root, and traveling down, the root file has subdirectories. Moreover, we can refer to particular types of directories with symbols as follows:

* ~ (tilde) to refer to the home directory.
* / (forward slash) also refers to home.
* . (dot) to refer to current.
* .. (dot dot) refers to the parent directory of the current one.

To provide the relative path of a file or a directory, we write it relative to our current working directory. While absolute paths relative to the root also contain the home symbol at the beginning.

Linux OS gives its users many advantages that other operating systems do not have.

* It deals with everything as a file, whether they are part of the software or even part of the machine's hardware, such as the monitor and the keyboard. Linux will treat them as if they were files.
* Linux will never ask you to provide file extensions. It can figure it out without the user's help. A special command you can use to tell the system to give you file types is "file [path]," with which it will determine the file type even if it was labeled with the wrong extension.
* Linux is case sensitive. So, we have to be careful with how we spell command and file paths.

Linux accepts spaces; however, it gets tricky very easily, so we have to use quotes and escape characters. Besides, hidden directories are easy to define and show with the unique "-a" flag, which could be provided as an argument to the "ls" command. We can also specify to hide a particular file or directory solely by prefixing the file's name with a dot (.) we can also rename it to make it hidden as such.