```
git log
```

- To view the commit history of a repository.
- A commit hash, which is a 40 character checksum data generated by SHA (Secure Hash Algorithm) algorithm. It is a unique number.
- Commit Author metadata, The information of authors such as author name and email.
- Commit Date metadata, It's a date timestamp for the time of the commit.
- Commit title/message, It is the overview of the commit given in the commit message.

```
git log --oneline
```

`--oneline` flag causes git log to display-

- One commit per line
- The first 7 characters of the SHA
- The commit message

```
 git log --stat
```

`--stat` flag causes git log to display-

- The modified files,
- The number of lines that have been added or removed
- A summary line of the total number of records changed
- The lines that have been added or removed.

```
git log --patch
git log --p
```

`--patch` flag is used to display

- Modified files
- The location of the lines that you added or removed
- Specific changes that have been made.

```
git log --graph
```

`--graph` flag is used to display

- ASCII graph representation of the commit history

```
git log --graph --oneline
```

`--graph --oneline` flag is used to display

- Compact representation of the commit history

```
git log --after="yy-mm-dd"
git log --after="21 days ago"
```

- Command will display all the commits made after the given date
- We can also pass the applicable reference statement like "yesterday," "1 week ago", "21 days ago," and more.

```
git log --after="2023-11-01" --before="2024-11-08"
```

- To track the commits that were created between two dates, pass a statement reference **--before** and **--after** the date

---

```
git log --author="Rabbil"
```

- Filter the commits by a particular user

```
git log --grep="Version 3"
```

- Filter the commits by the commit message.