File permissions

Every file in linux has the following 3 permissions associated:

- Read - We can only read a file
- Write - We can edit a file content
- Execute - We can execute the file

We can make a file get any combination of permissions i.e. READ WRITE, or READ EXECUTE, OR WRITE, OR EXECUTE, OR READ WRITE EXECUTE and so on.
So a file can may be have all these 3 permissions or a subset of them.

These permissions are associated on different level of ownership. Ownership is based on user level, that it includes:

- users who owns the file
- user group that owns the file
- other users who has access to the system
- all the users of the system.

For any user of the machine we can separately define what permissions they have, like owner of the file can Read write execute and may be others can only read.

chmod

chmod command helps us to change permission of a file or directory.
There are two ways to use the chmod command:

1. Symbolic way

```Shell
chmod [user][operator][permission] filename
```

- user: User part describes which set of users who want this chmod command to affect ? It can have one of the following values:
  - **u**: Owner
  - **g**: Group
  - **o**: Others
  - **a**: all users (owner group and others)
- operator: This operator tells if we want to add a permission or remove a permission or just set an exact permission by removing other. It can have the following values:
  - **+**: It will be adding a permission
  - **-**: It will be removing a permission
  - **=**: It will be allocating an exact permission and removing everything else
- permission: The permission we want to associate.
  - **r**:read
  - **w**:write
  - **x**:execute

Examples:

If we want to add execute permission for the owner of the file

```bash
chmod u+x script.sh
```

If we want to remove write permission for the group

```bash
chmod g-w script.sh
```

2. Numeric way

Every file gets permission associated to it in a group of 3 characters for owner group and others.
Ex: `rwxr-wr--`
The first `rwx` shows owner of the file has read write and execute access, group has read and execute access and other only can read the file.

We can represent the 3 characters associated to owner / group / other in the form of a number in the range [0-7]. How ? Because the numbers from 0-7 can be represented in the form of a 3 digit binary.
Ex: 7: 111
Ex: 5: 101
Ex: 3: 011

So if we have a 3 digit binary, every character of binary shows whether we have that permission or not. Ex: 4 : 100 , so we have only read permission. Ex: 5 : 101, means we have read and execute permission.

So, for each user, group and other we can put a decimal number in the range 0-7 and change permission

```shell
chmod 444 script.sh
```

4 means 100 in binary which is only read permission, so owner group and other all get read access only.

```shell
chmod 400 script.sh
```

Now only owner has read access else no one can do anything.

```shell
chmod 700 scrip.sh
```

Owner of the file has read write execute and no body else has anything because 7 in binary is 111.

chown

This command can change the ownership of the file i.e. who the file belongs to.

```bash
chown [owner][:group] file
```