Create file(s)

In order to create empty file in Linux, we have to use touch command. Let's do it.

touch testfile

Do we have it?

ls -l testfile

Yes, the file exists. We also can see that the file has 0 size, another words, it is empty.

Of course, like in previous scenario, we can create multiple files.

touch my{01..100}file

Let's use the wildcard character \* to list all of them

ls my\*file

It is not what we really can use here, but if \* means any string, sometimes we need to list specific subgroup. Let's prepare ourselves first.

touch try1 try2 try01

Ok, now, we want to list try1 and try2 only. If I use

ls try\*

we will receive all three files.

Instead of \* we can use ?. This sign means any single character.

Let's try

ls try?

Yep, this is it.

Let's now create some structure.

mkdir testdir  
touch testdir/mytest{01..1000} testdir/file{01..1000}

Do you know what we've done?

# Delete file(s)

We already know the proper command. It is rm.

We can remove single file

rm try01

We can remove more than one file

rm try1 try2

We obviously can remove multiple files

rm testdir/mytest{01..1000}

And, yes, we can remove all files, directories, the whole structure

rm -rf testdir

What -rf means? It is explained in previous lesson.

# Vim

Ok, vim is not part of this lesson. But I'd like to make sure, that we understand that touch is not only one way to create files.

Let's run

vim mynewfile

We opened a vim editor. We wil lhave lessone about it later, now it is enough to follow the instruction.

In order to start typing anything in editor (enter insert mode), hit

i .

Put some text, for example

this is my great line

Now, hit ESC key to enter the command mode, and type :wq

You left vim and now you can check the file

ls -l mynewfile

Yes, we have the file and it is not empty!