

## MODULO 9

Uno script è una sequenza di N comandi concatenati tra di loro scritti all'interno di un file di testo.

I comandi sono eseguiti uno dopo l'altro, in modo sequenziale, dando così vita ad una piccola applicazione e permettendo l'automatizzazione.

L'estensione associata agli script è .sh

Usiamo Nano per creare/modificare il file di script

Come primo comando (prima riga del editor di testo): `#!/bin/bash`

- dice all'OS di usare la bash per eseguire lo script
- `#!/bin/bash` viene chiamato shebang

```
GNU nano 6.2 script1.sh
#!/bin/bash

mkdir backup
cp dati.txt /backup
chmod 444 backup/dati.txt

cate@cate-virtual-machine:~$ ls
dati.txt Desktop destinazione Documents Downloads Music origine Pictures Public script.sh snap Templates Videos
cate@cate-virtual-machine:~$ touch script1.sh
cate@cate-virtual-machine:~$ ls
dati.txt Desktop destinazione Documents Downloads Music origine Pictures Public script1.sh script.sh snap Templates Videos
cate@cate-virtual-machine:~$ nano script1.sh
cate@cate-virtual-machine:~$ ls -l
total 56
-rw-rw-r-- 1 cate cate 28 ott 17 17:00 dati.txt
drwxr-xr-x 2 cate cate 4096 feb 21 2024 Desktop
drwxrwxr-x 2 cate cate 4096 ott 18 10:35 destinazione
drwxr-xr-x 2 cate cate 4096 ott 15 08:21 Documents
drwxr-xr-x 2 cate cate 4096 feb 21 2024 Downloads
drwxr-xr-x 2 cate cate 4096 feb 21 2024 Music
drwxrwxr-x 2 cate cate 4096 ott 17 16:42 origine
drwxr-xr-x 2 cate cate 4096 feb 21 2024 Pictures
drwxr-xr-x 2 cate cate 4096 feb 21 2024 Public
-rw-rw-r-- 1 cate cate 72 ott 18 10:37 script1.sh
-rwxr-xr-- 1 cate cate 602 ott 16 15:53 script.sh
drwx----- 3 cate cate 4096 feb 21 2024 snap
drwxr-xr-x 2 cate cate 4096 feb 21 2024 Templates
drwxr-xr-x 2 cate cate 4096 feb 21 2024 Videos
cate@cate-virtual-machine:~$ chmod a+x script1.sh
cate@cate-virtual-machine:~$ ls -l script1.sh
-rwxrwxr-x 1 cate cate 72 ott 18 10:37 script1.sh
cate@cate-virtual-machine:~$ ./script1.sh
```

Questo non funziona (usato per dimostrare che lo script continua ad eseguire i comandi e se trova un errore (non logico) ti stampa l'errore ma continua con l'esecuzione:

```
GNU nano 6.2 script1.sh *
#!/bin/bash

DATA=$(date+%Y-%m-%d)
mkdir backup_$DATA
```

Versione corretta:

```
GNU nano 6.2 script1.sh
#!/bin/bash

DATA=$(date +"%Y-%m-%d")
mkdir backup_$DATA
cp dati.txt backup_$DATA/
```

```

GNU nano 6.2 script1.sh *
#!/bin/bash

echo "Inserisci il nome del file da creare"
read nome_file
touch $nome_file
echo "File $nome_file creato!"

```

```

GNU nano 6.2 ./script1.sh
#!/bin/bash

echo "Inserisci il nome del file da creare"
read nome_file
touch $nome_file e;2> /dev/null
echo "File $nome_file creato!"

```

```

GNU nano 6.2 script1.sh
#!/bin/bash

if [ -f dati.txt ]; then
echo "Il file dati.txt esiste."
else
echo "Il file dati.txt non esiste."
fi

```

Type	Symbol	Example
True if file is a directory	-d	-d file
True if file is a plain file	-f •	-f file
True if file exists	-e	-e file
True if file has read permission for current user	-r	-r file
True if file has write permission for current user	-w	-w file
True if file has execute permission for current user	-x	-x file

Type	Symbol	Example
True if length of string is zero	-z	test - "string"
True if length of string is not zero	-n	test -n "string"
True if strings are equal	=	String1 = string2
True if strings are not equal	!=	String1 != string2
True if integers are equal	-eq	Int1 -eq int2
True if integers are not equal	-ne	Int1 -ne int2
True if first integer is greater than second integer	-gt	Int1 -gt int2
True if first integer is greater than or equal to second integer	-ge	Int1 -ge int2
True if first integer is less than to second integer	-lt	Int1 -lt int2
True if first integer is less than or equal to second integer	-le	Int1 -le int2