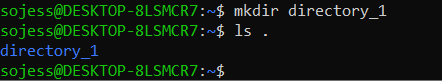
**LINUX/UNIX**

**mkdir**

**mkdir directory\_name** : It creates a directory of name directory\_name

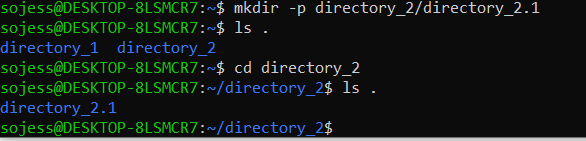
**ls .** :lists all the directories in the current folder

Eg:



**mkdir -p directory\_name1/directory\_**name2 : creates nested directory

Eg:



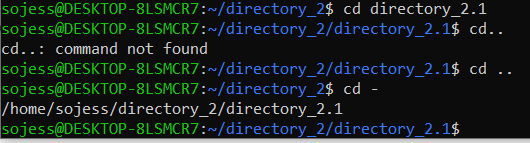
**cd**

**cd directory\_name** :takes you to the directory directory\_name

cd .. : takes you one directory before

cd - : takes you to previous directory

eg:



**touch**

touch file\_name: touch is used to create a file of file name file\_name

eg:



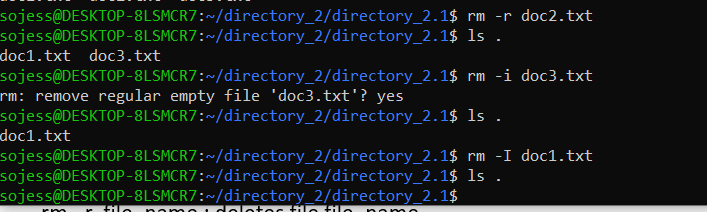
**rm**

**rm –r file\_name** : deletes file file\_name

**rm -i file\_name**: ask user permission before deletion

**rm -I file\_name** : doesn’t ask user permission before deletion

eg:

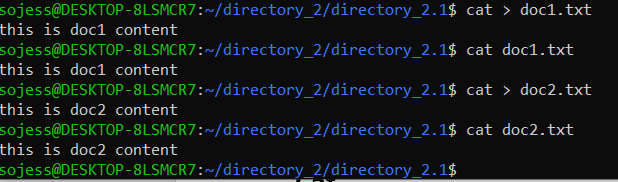


**cat**

**cat > file\_name** : This command creates a new file file1 (in write mode) if it doesn’t exist in the present working directory. If any file with file name file1 exists in the current directory then it is overwritten. After writing the text into the file, press ctrl+d to save and exit from the writing mode.

**cat file\_name** : this reads the file file\_name

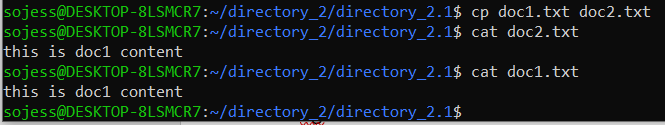
eg:



**cp &mv**

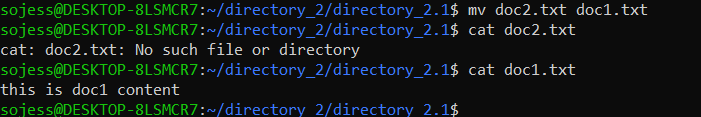
**cp parent\_file child\_file**: copy content of parent\_file to child\_file. The existing content in child\_file is overwritten

eg:



**mv parent\_file child\_file**: move content of parent\_file to child\_file And the parent file is removed

eg:



**echo**

echo “text” : prints text on screen

eg:



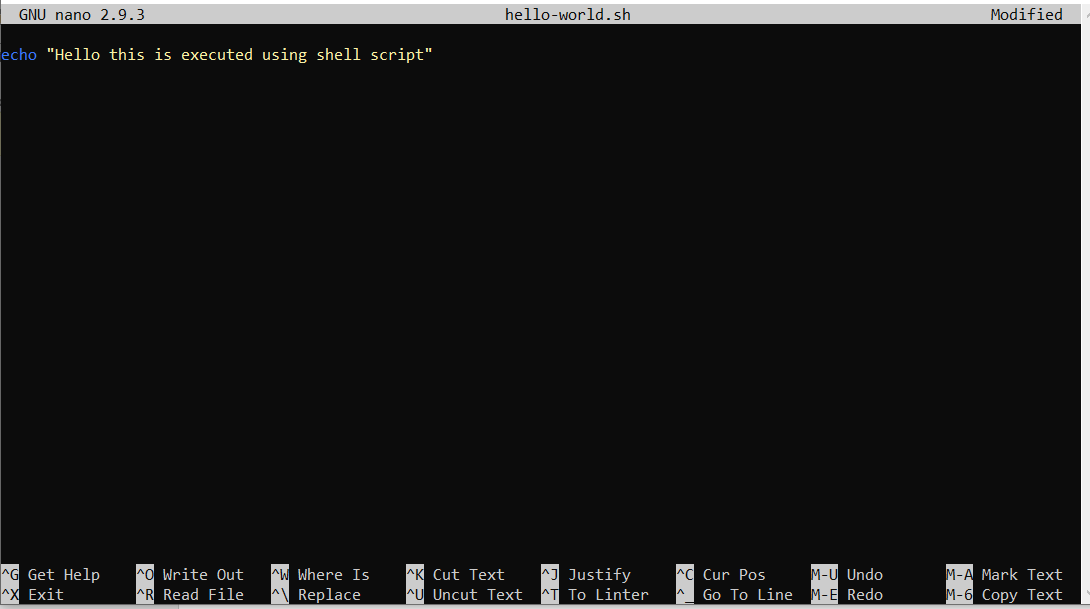
**Shell scripting**

**nano**

**nano file\_name.sh :** creates a shell script , enter the script and press ctrl+x to save and exit

eg:





**chmod +124 file\_name.sh** or **chmod a+x file\_name.sh**:Change the file permission to executable

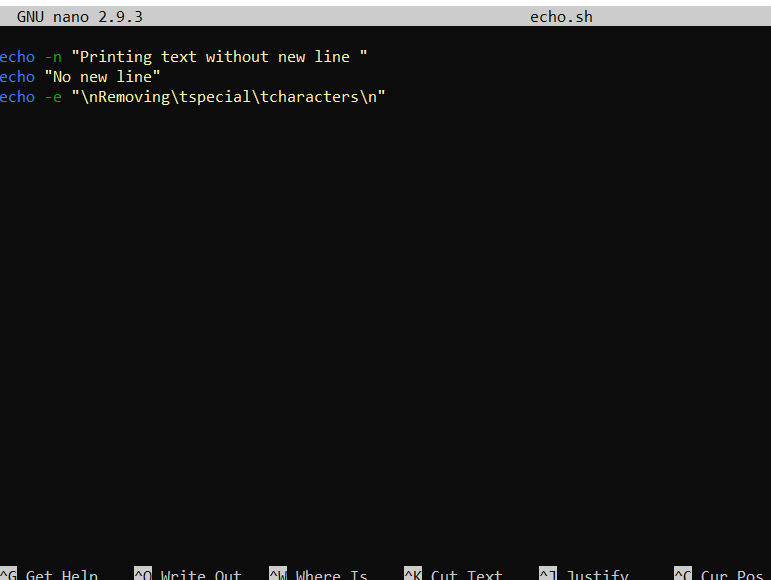


**./file\_name.sh** or **bash file\_name.sh** :this executes the file file\_name.sh

Eg:

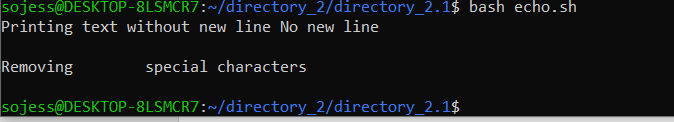


**Echo script**

****

echo -n : print without new line

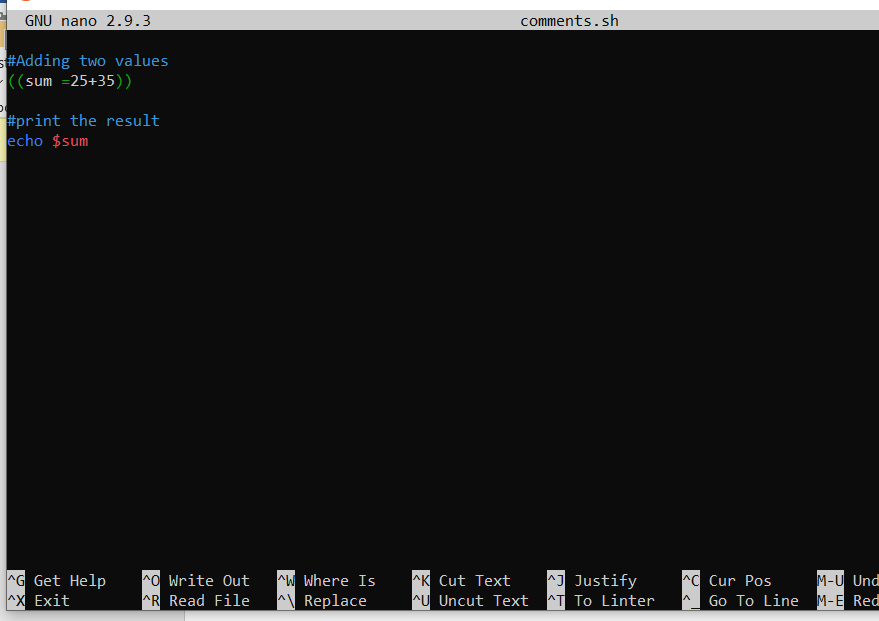
echo -e: e tells echo that the string passed to it contains special characters



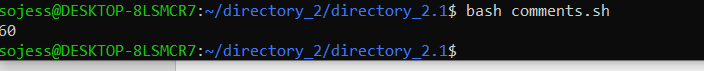
**Comments**

**Single line:**

Comments are useful for documentation and are a requirement for high-quality codebases. It’s a common practice to put comments inside codes that deal with critical logic. To comment out a line, just use the **#**(hash) character before it

****

Output:



**While loop**

Syntax:

while [ condition ]

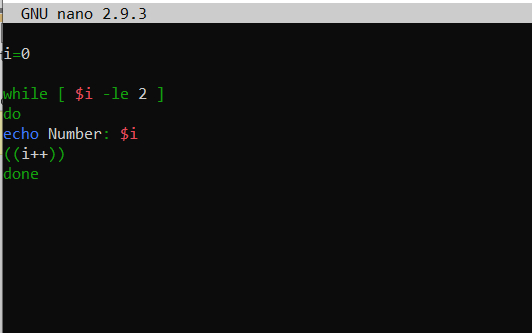
do

commands 1

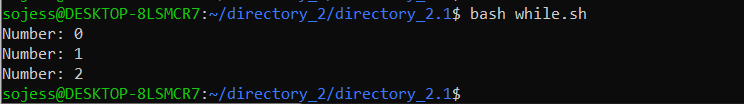
commands n

done

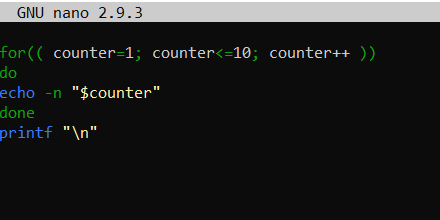
eg:



Output:



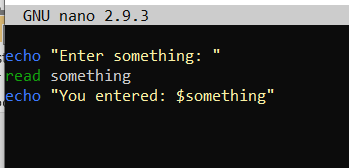
**For loop**



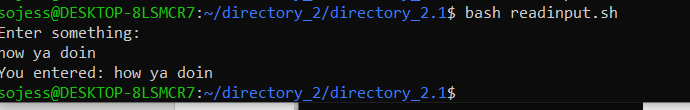
Output:



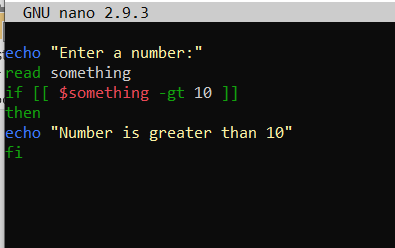
**Read Input**



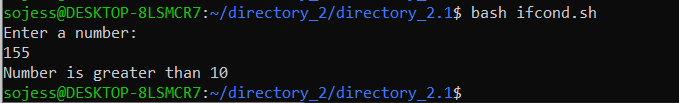
Output:



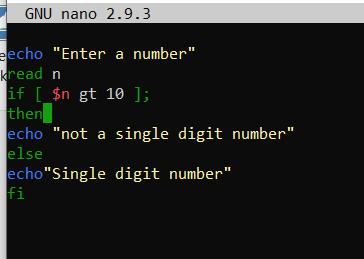
**If condition**



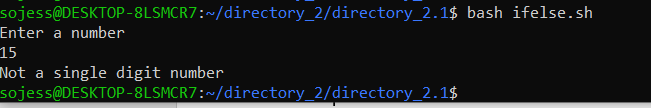
Output:



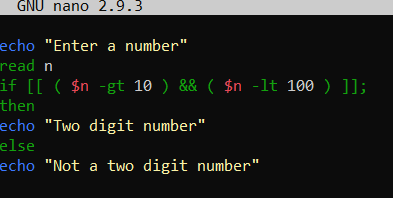
**If else**

****

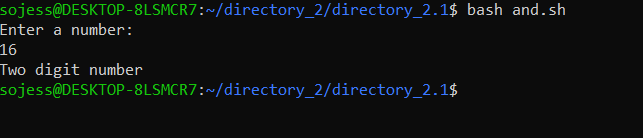
Output:



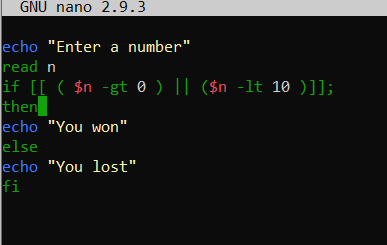
**AND operator**



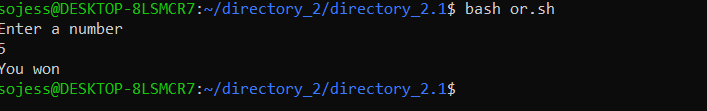
Output:



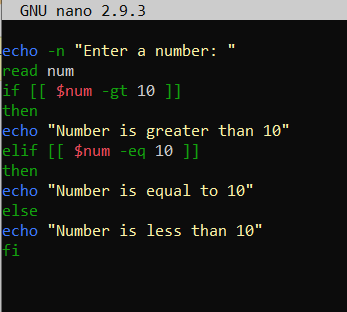
**OR operator**



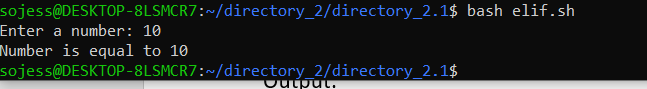
Output:



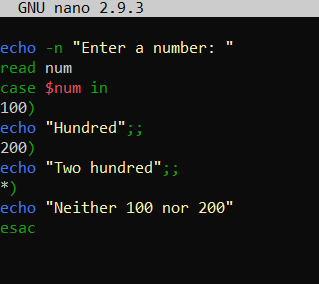
**Elif**



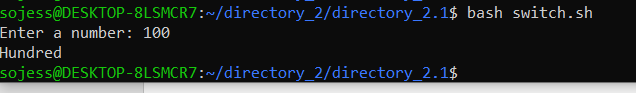
Output:



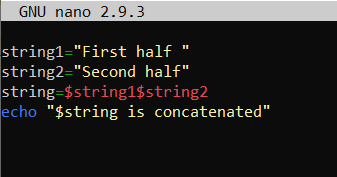
**Switch**



Output:



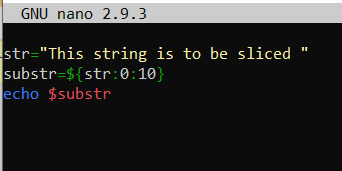
**String concatenation**



Output:



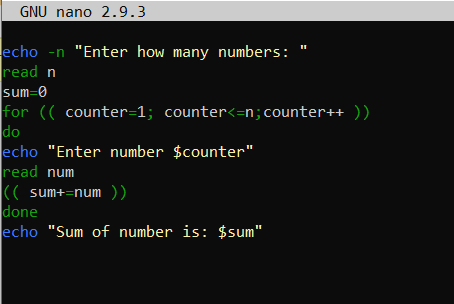
**String slicing**



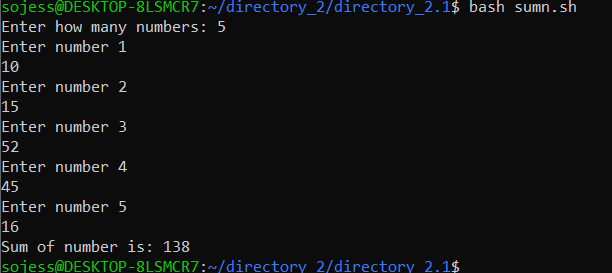
Output:



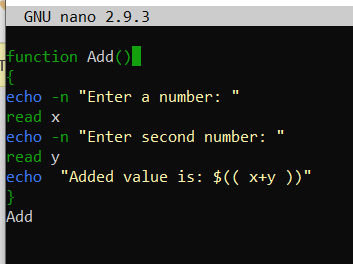
**Sum of n numbers**



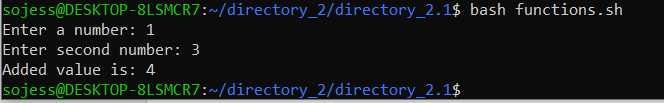
Output:



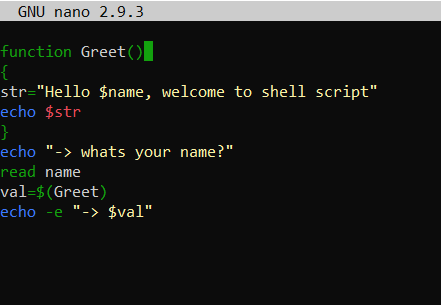
**Functions**



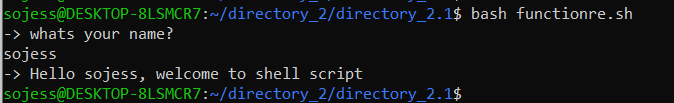
Output:



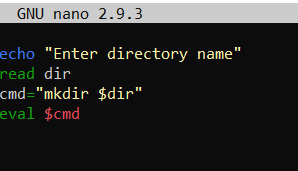
**Functions with Return Values**



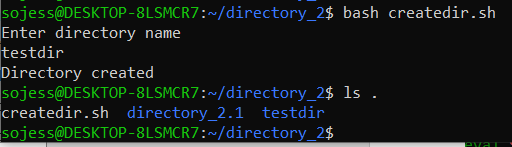
Output:



#### ****Creating Directories from Bash Scripts****



Output:



#### ****Create a Directory after Confirming Existence****

#### 

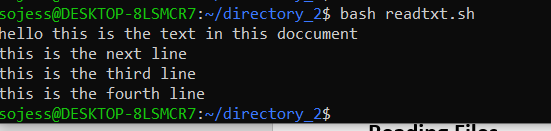
#### Output:

#### 

#### ****Reading Files****

#### 

Output:



#### ****Appending to Files****

#### 

#### Output:

#### 

#### ****Parsing Date and Time****

#### 

#### Output: