[1]-shell script
procedure:
----

step 1:create .sh file
```
 >>touch hello.sh
```

step 2:go into the .sh file:
```
 >>gedit hello.sh
```

step 3:shell type declare into .sh file(good practice):
```
 #!/bin/bash
```

step 4:write shell commands require:
```
 example:
  echo hello world

  x=2

  echo this is variable x : $x
  cooment in shell script:
  -----------------------
  #(------)
```

```
-----------------------------------------------
|file permission:    |
| read(r)     |
| write(w)    |
| execute(x)    |
|      |
|   |---||---||---|  |
| d/-[file/folder] |owner||group||other| |
-----------------------------------------------
```

step 5:give permission if needed:
```
 >>chmod +x hello.sh
```

step 6:run script file(.sh):
```
 >>./hello.sh
```

----

[1]-shell script format
[1]-execute a .sh file

----

[2]-variable:

```
 -system variable
  all-capital letter
 -user variable
  all-small letter

 system variable print:
  echo $USER #user name
  echo $PWD #current working dir
  echo $HOME #home directory
  echo $BASH #shell name

  name=mark
  echo $name
```

-in variable declare space forbidden

```
 x="hello"
 y=hello

 echo $x $y[variable declare works either way ]
```

types of echo in shell script:

```
  myvar=hello
  echo $myvar # print variable
  echo "$myvar" # print variable
  echo '$myvar' # print variable name only
  echo \$myvar # print variable name only
```

[3]-user input(read) in shell script:

```
read name
echo "my name is : " $name
```

message print before input:

```
 read -p 'enter your name : ' name
 echo "my name is : " $name
```

keep input hidden:

```
 read -p 'enter your name : ' name
 read -p 'enter your id : ' std_id
 read -p 'enter password : ' -s pass #-s for keep input hidden
 echo "my name is : " $name
 echo "student id : " $std_id
 echo "password : " $pass
```

if no variable decalre data can be store in system variable $REPLY:

```
 read -p "enter names : "
 echo "the value is : " $REPLY
```

[4]-array in shell script:

```
#names: max , john , mark

read -p "enter names : " -a names #-a for array
echo "names[0] :" ${names[0]} #${variable[position]}
echo "names[1] :" ${names[1]} #${variable[position]}
echo "names[2] :" ${names[2]} #${variable[position]}
```

[ ]-commad line argument in shell script[direct in command input]:

```
>>./hello.sh max jhon mark
echo "first cla variable : $1"
```

```
 echo "second cla variable : $2"
 echo "third cla variable : $3"
 #echo "0 cla variable : $0" #shell script name stored
```

[5]-array in command input(command line argument):

```
 #./hello.sh max jhon mark
 #args={"max","jhon","mark"}
 #shell script array index starts from 0

 args=("$@")

 echo "first cla variable : ${args[0]}"
 echo "second cla variable : ${args[1]}"
 echo "third cla variable : ${args[2]}"

 echo "number of arguments : " $#
```

[6]-expression:

```
 #x=1+1 #x=2
 #y=21-2 #y19


 #expr :

 x=$(expr 1 + 1) #space between values & expression
 echo $x

 num1=20
 num2=5

 result=$(expr $num1 + $num2)
 echo $result
```

[7]-if else....

[8]-case