1. # The following line won't run because of a syntax error
   print("hi)

```
File  Actions  Edit  View  Help
#!usr/bin/python
print("hi)
~
```

Output :

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
  File "/home/kali/lab1.py", line 2
    print("hi)
          ^
SyntaxError: unterminated string literal (detected at line 2)
```

Corrected code:

```
File  Actions  Edit  View  Help
#!usr/bin/python
print("hi")
~
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
hi
```

2. # Exercise 2
''' The following lines won't run properly,
   even if the syntax error in the line above is corrected,
   because of a run-time error '''
print(hello)

```
File  Actions  Edit  View  Help
#!usr/bin/python
print(hello)
~
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
Traceback (most recent call last):
  File "/home/kali/lab1.py", line 2, in <module>
    print(hello)
          ^^^^^
NameError: name 'hello' is not defined. Did you mean: 'help'?
```

Corrected Code:

```
File  Actions  Edit  View  Help
#!usr/bin/python
print("hello")
~
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
hello
```

3. # Display a string (greeting message) directly

```
┌──(kali㉿kali)-[~]
└─$ echo "Hello World"
Hello World
```

4. # Display the contents of a string variable

```
File  Actions  Edit  View  Help
#!usr/bin/python
s="Hey There!!!"
print(s)
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
Hey There!!!
```

5. # Display the string which contains single quotes
   Ex: Indian's

```
┌──(kali㉿kali)-[~]
└─$ echo "Indian's"
Indian's
```

6. # Display the string which contains Double Quotes
   Ex: Students,"Welcome to SOIS".

```
┌──(kali㉿kali)-[~]
└─$ echo "Students,\"Welcome to SOIS\""
Students,"Welcome to SOIS"
```

6. Read two numbers in (user input) and store as num1 and num2, Calculate the sum, difference, product, Quotient, reminder, power

```
#!usr/bin/python
num1=int(input())
num2=int(input())
print(f"The Sum of {num1} and {num2} is {num1+num2}")
print(f"The Difference of {num1} and {num2} is {num1-num2}")
print(f"The Product of {num1} and {num2} is {num1*num2}")
print(f"The Quotient of {num1} and {num2} is {num1//num2}")
print(f"The Reminder of {num1} and {num2} is {num1%num2}")
print(f"The Power of {num1} and {num2} is {num1**num2}")
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
10
5
The Sum of 10 and 5 is 15
The Difference of 10 and 5 is 5
The Product of 10 and 5 is 50
The Quotient of 10 and 5 is 2
The Reminder of 10 and 5 is 0
The Power of 10 and 5 is 100000
```

7. check the value of num1 is integer or not?

```
#!usr/bin/python
num1=input()
num2=input()

if num1.isdigit():
    print("Yes, the given input value is Integer")
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
2
3
Yes, the given input value is Integer
```

8. convert into integer

```
#!usr/bin/python
num1=input()
num2=input()

print(f"Converting the given input value into Integer = {int(num1)}")
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
1233
234
Converting the given input value into Integer = 1233
```

9. Find the datatype for the variable num1 and num2.

```
#!usr/bin/python
num1=input()
num2=input()

print(type(num1))
print(type(num2))
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
122
12234
<class 'str'>
<class 'str'>
```

10. read the float value from the user and print the number rounded to 2 decimal places

```
#!usr/bin/python
num1=float(input())
num2=input()

print(round(num1,2))
~
~
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
23.345
23
23.34
```

11. read the float value from the user and print the absolute value

```
#!usr/bin/python
num1=float(input())
num2=input()

print(abs(num1))
~
```

t

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
223.2334
23
223.2334
```

12. Store different type values in the variabale
    String
    numeric
    complex
    list
    dictionary
    set
    tuple

```
#!usr/bin/python
string="HI"
n=10
cmplx=10+4j
l=[1,2,3]
d={'1':1,'2':1,'3':1}
s=set([1,2,3])
t=(1,2,3)
~
```

13. Find the data type for the above variables

```
#!usr/bin/python
string="HI"
n=10
cmplx=10+4j
l=[1,2,3]
d={'1':1,'2':1,'3':1}
s=set([1,2,3])
t=(1,2,3)
print(type(string))
print(type(n))
print(type(cmplx))
print(type(l))
print(type(d))
print(type(s))
print(type(t))
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
<class 'str'>
<class 'int'>
<class 'complex'>
<class 'list'>
<class 'dict'>
<class 'set'>
<class 'tuple'>
```

14. # Display the number of letters in the string
greeting = "Welcome to Python Programming"

```
#!usr/bin/python
greeting="Welcome to Python Programming"
print(len(greeting))
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
29
```

15. read the first name and last name from the user and combine first name and last name. combine name and greeting message

```
#!usr/bin/python
greeting="Welcome to Python Programming"
firstname=input()
lastname=input()

print(firstname+lastname+" " +greeting)
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
Surya
Devi
SuryaDevi Welcome to Python Programming
```

16. Display the string with space
    Ex: firstname lastname

```
#!usr/bin/python
firstname=input()
lastname=input()

print(firstname+" "+lastname)
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
Surya
Devi
Surya Devi
```

17. Display first two characters from the name

```
#!usr/bin/python
firstname=input()
lastname=input()
name=firstname+lastname
print(name[:2])
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
Surya
Devi
Su
```

18. Display last three characters from the name

```
#!usr/bin/python
firstname=input()
lastname=input()
name=firstname+lastname
print(name[-3:])
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
Surya
Devi
evi
```

19. Display 3rd character to last character

```
#!usr/bin/python
firstname=input()
lastname=input()
name=firstname+lastname
print(name[-3])
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
Surya
Devi
e
```

20. Display 3rd to 5th character

```
#!usr/bin/python
firstname=input()
lastname=input()
name=firstname+lastname
print(name[2:6])
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
Surya
Devi
rya
```

21. Create a list of food with two elements.

```
#!usr/bin/python
food=["dosa","idly"]
```

22. Add one more to the food list using .append()

```
#!usr/bin/python
food=["dosa","idly"]
food.append("pongal")
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
['dosa', 'idly', 'pongal']
```

23. Add two more food strings to food using .extend()

```
#!usr/bin/python
food=["dosa","idly"]
food.append("pongal")
food.extend(["poori","chapati"])
print(food)
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
['dosa', 'idly', 'pongal', 'poori', 'chapati']
```

24. Count total number of items in the list

```
#!usr/bin/python
food=["dosa","idly"]
food.append("pongal")
food.extend(["poori","chapati"])
print(len(food))
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
5
```

25. Print the first two items in food using slicing notation

```
#!usr/bin/python
food=["dosa","idly"]
food.append("pongal")
food.extend(["poori","chapati"])
print(food[:2])
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
['dosa', 'idly']
```

26. Print the last item in food using index notation

```
#!usr/bin/python
food=["dosa","idly"]
food.append("pongal")
food.extend(["poori","chapati"])
print(food[-1])
```

Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
chapati
```

27. Debug: Program is to check the given number is odd or even

```
number = input("Enter a number: ")
x = str(number)/2
if x == 0
   print("The number is Even.")
else
   print("The number is Odd.")
```

```
#!usr/bin/python

number =input("Enter a number: ")
x =int(number)%2
if x == 0:
    print("The number is Even.")
else:
    print("The number is Odd.")
```

28. Debug: Program is to convert centigrade to Fahrenheit

```
c = input("Enter temperature in Centigrade: ")
f = 9*(int(c)/5 +32
print("Temperature in Fahrenheit is: ", f)
```

```
#!usr/bin/python

c = input("Enter temperature in Centigrade: ")
f = int(c)*(9/5) +32
print("Temperature in Fahrenheit is: ", f)
```

29. Debug:

```
int = int(input("Enter the count of numbers: "))
i = 0
summ= 0
for i in range(count):
    x = int(input("Enter an integer: "))
    sum = sum + x
    avg = sum/count

print("The average is: ", avg)
```

```
#!usr/bin/python
count = int(input("Enter the count of numbers: "))
i = 0
summ= 0
for i in range(count):
    x = int(input("Enter an integer: "))
    summ = summ + x
    avg = summ/count

print("The average is: ", avg)
```

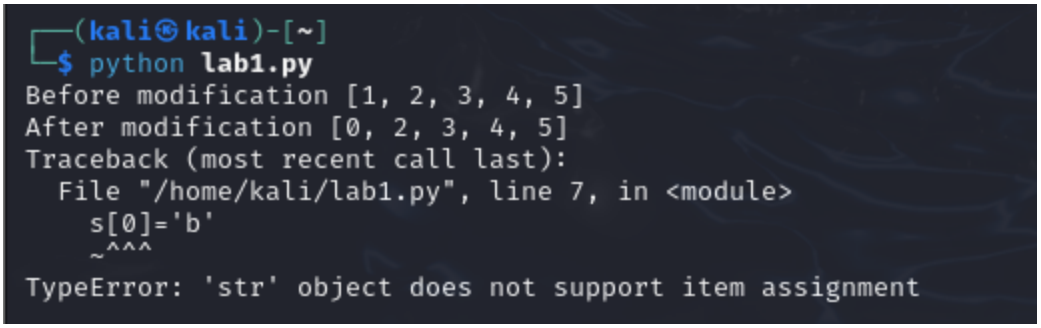
Output:

```
┌──(kali㉿kali)-[~]
└─$ python lab1.py
Enter the count of numbers: 5
Enter an integer: 1
Enter an integer: 2
Enter an integer: 3
Enter an integer: 4
Enter an integer: 5
The average is:  3.0
```

30. Prove : strings is not mutable
lists are mutable

```
#!usr/bin/python
s="Hello"
l=[1,2,3,4,5]
print("Before modification",l)
l[0]=0
print("After modification",l)
s[0]='b'
```

Output:



==========
==
Deadline 14.08.2024
==========
==