S.H.E.L.L.

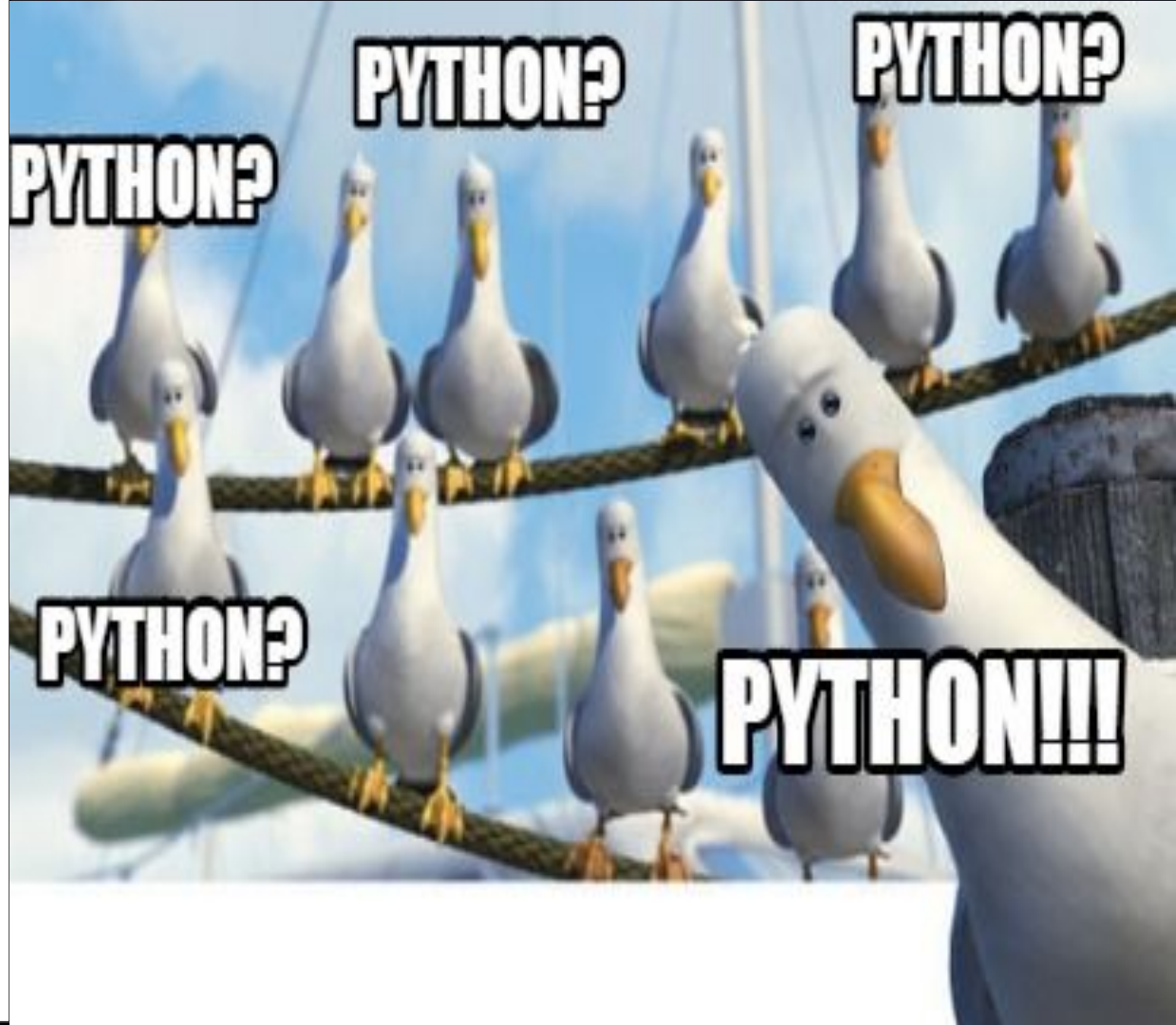
SECURITY IS AN ILLUSION

What do you know about python?

- It is a high level programming language and Object Oriented scripting language.
- It doesn’t not have format specifiers unlike C language which do need them.
- Indentation is very important in python.
- It is a interpreting language.
- It has vast collection of libraries and packages.

S.H.E.L.L.
SECURITY IS AN ILLUSION

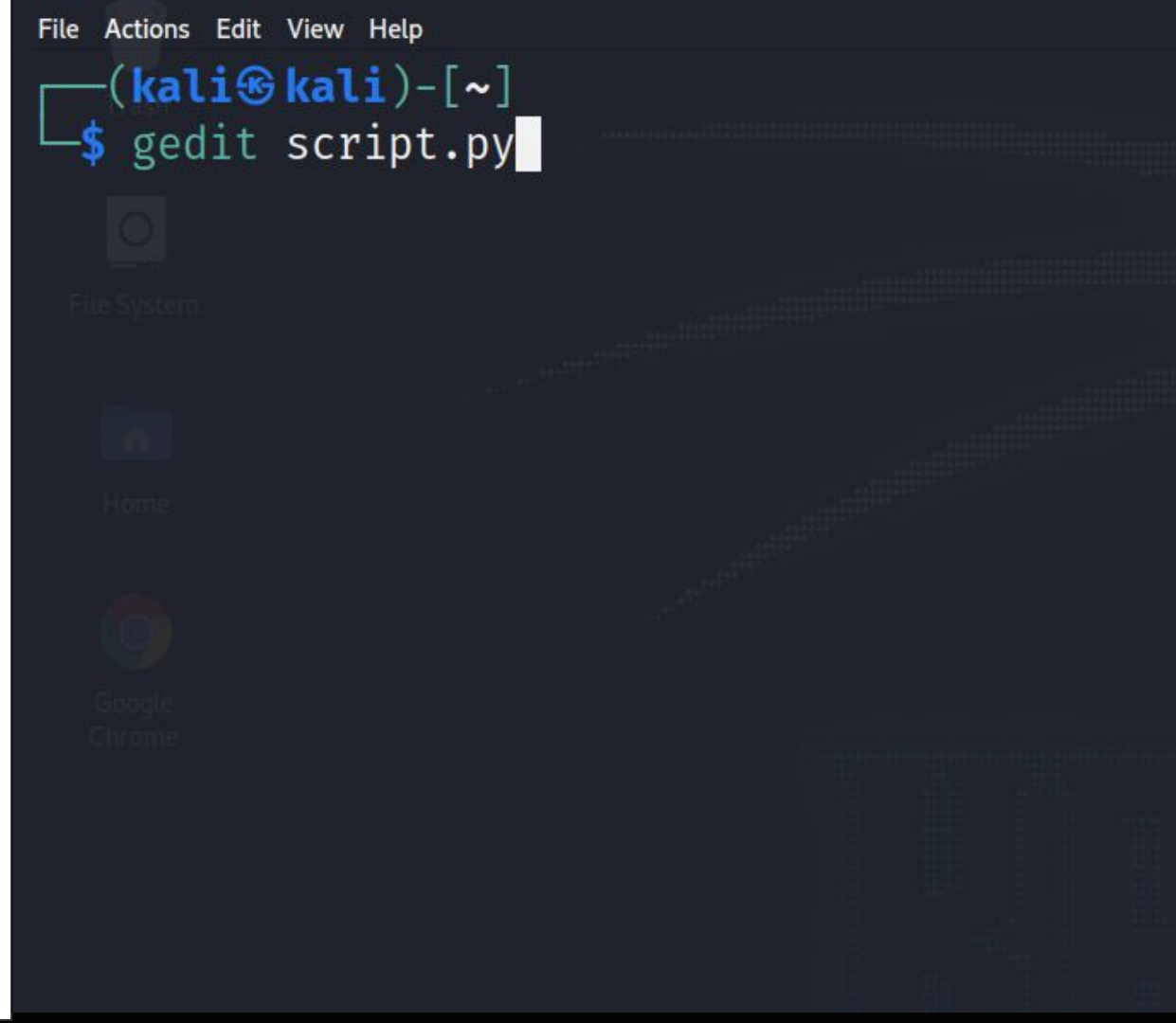
How to start coding in python?



Installing gedit

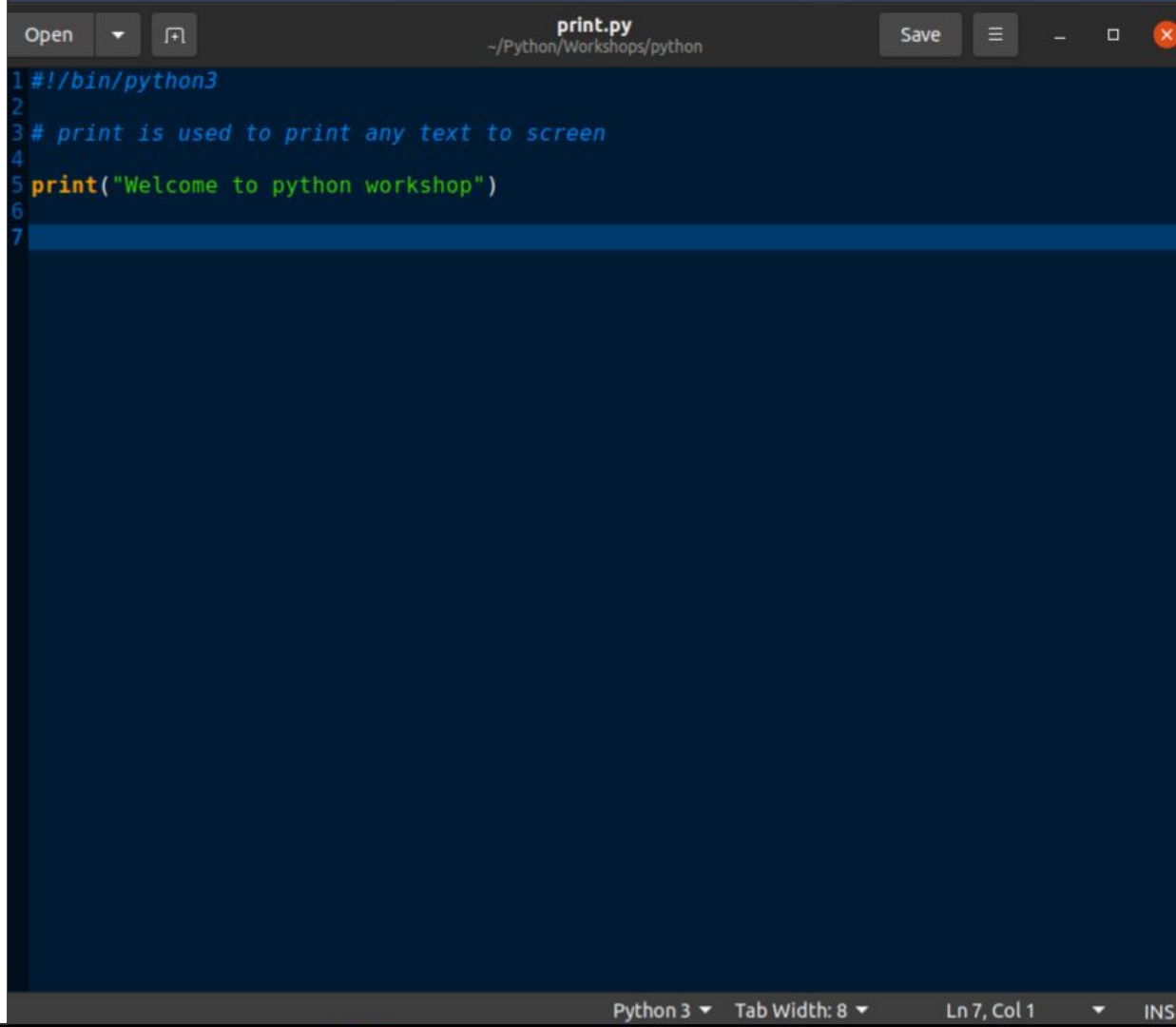
```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install gedit
```

Beginning to code in python



S.H.E.L.L.
SECURITY IS AN ILLUSION

Print Statement



Output Statement

lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Workshops/python

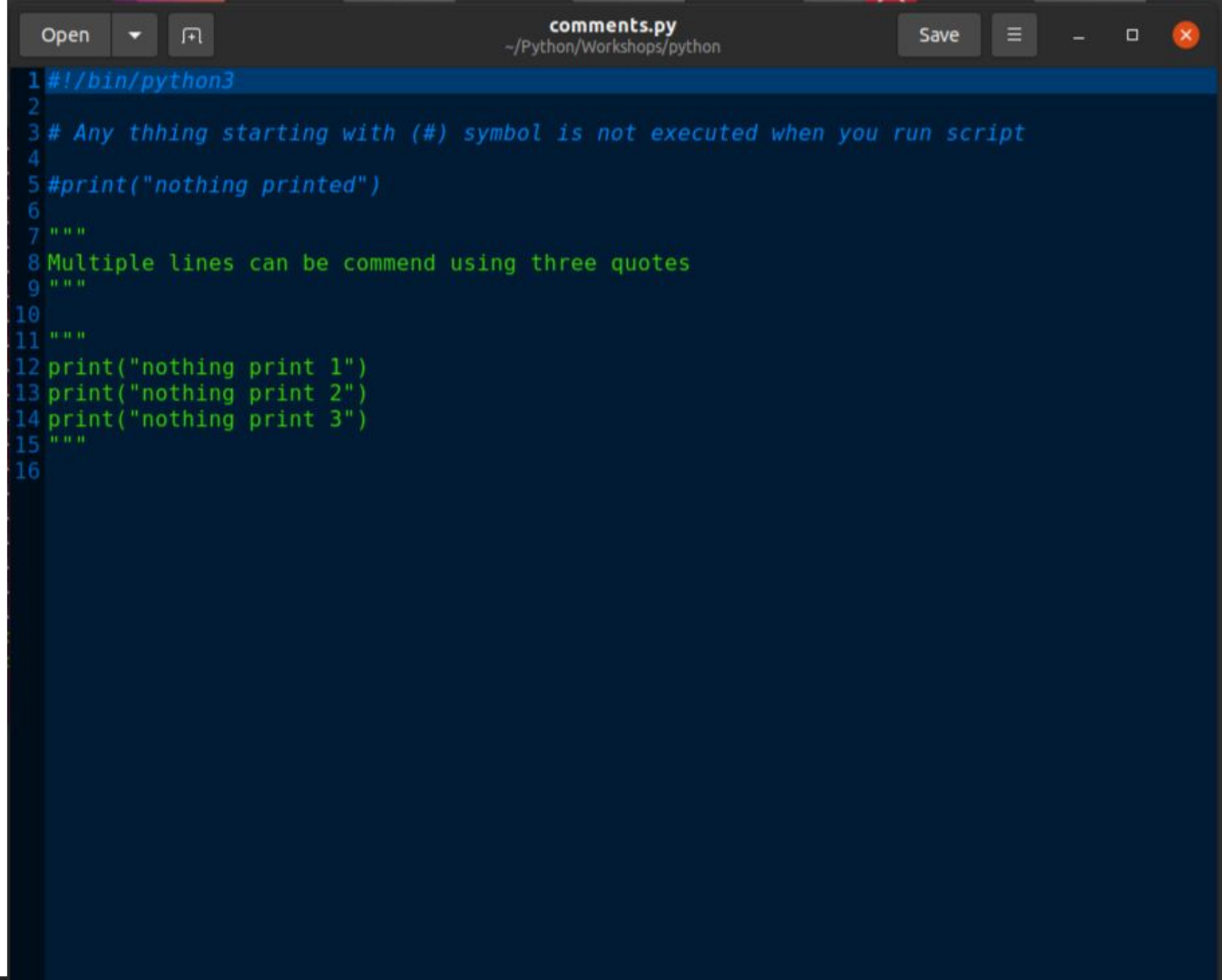
lelouch@lelouch-IdeaPad-3-15IML05-D: ~/P...

lelouch@lelouch-IdeaPad-3-15IML05-D: ~/P...

```
lelouch@lelouch-IdeaPad-3-15IML05-D:~/Python/Workshops/python$ python3 print.py
Welcome to python workshop
lelouch@lelouch-IdeaPad-3-15IML05-D:~/Python/Workshops/python$
```

S.H.E.L.L.
SECURITY IS AN ILLUSION

How to Comment



S.H.E.L.L.
SECURITY IS AN ILLUSION

Variables

variable.py
~/Python/Workshops/python

```
#! /usr/bin/python3


x = 5           # integer variable
y = "SHELL"     #string variable
z = 123.69      #Float variable
a = True        #Boolean variable (Values are True or False)
b = 'a'         #Character data-type (string with length 1)

#printing the values of the variables
print('x = ',x)
print('y = ',y)
print('z = ',z)
print('a = ',a)
print('b = ',b)

#In order to find the data type of the variable,we use 'type'.
print("\n") # \n is a new line character
print("Data-type of x is",type(x))
print("Data-type of x is",type(y))
print("Data-type of x is",type(z))
print("Data-type of x is",type(a))
print("Data-type of x is",type(b))
```

Variable output

```
lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Workshops/python
lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Works...
lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Works...
lelouch@lelouch-IdeaPad-3-15IML05-D:~/Python/Workshops/python$ python3 variable.py
x =  5
y =  SHELL
z =  123.69
a =  True
b =  a


Data-type of x is <class 'int'>
Data-type of x is <class 'str'>
Data-type of x is <class 'float'>
Data-type of x is <class 'bool'>
Data-type of x is <class 'str'>
lelouch@lelouch-IdeaPad-3-15IML05-D:~/Python/Workshops/python$
```

Operators

Open | mathematical_operation.py ~/Python/Workshops/python | Save

```python
x = 15
y = 4

print("x = ", x, "and y = ", y)
print("\n")

# addition
z = x + y
print("Addition x+y =",z)  # print(x+y)
print("\n")

# subtraction
z = x - y
print("Subtraction x-y =",z)
print("\n")

# multiplication
z = x * y
print("Multiplication x*y =",z)
print("\n")

# division
z = x / y
print("Division x/y =",z)
print("\n")

# quotient
z = x // y
print("Quotient x//y =",z)
print("\n")

# remainder
z = x % y
print("Remainder x%y =",z)
print("\n")

# increment
z = x
print("Initial z =",z)
z = z + 1
z +=1           #short hand
print("Twice incremented z =",z)

print("""\n similarly you can short hand other operations like :
                i = i - 2  ----->  i -= 2
                i = i / 2  -----> i /= 2
                i = i * 2  ----->  i *= 2
        """)
```

Operators

lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Workshops/python

lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Works... | lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Works...

```
lelouch@lelouch-IdeaPad-3-15IML05-D:~/Python/Workshops/python$ python3 mathematical_operation.py
x =  15 and y =  4


Addition x+y = 19


Subtraction x-y = 11


Multiplication x*y = 60


Division x/y = 3.75


Quotient x//y = 3


Remainder x%y = 3


Initial z = 15
Twice incremented z = 17

 similarly you can short hand other operations like :
                i = i - 2  ----->  i -= 2
                i = i / 2  ------> i /= 2
                i = i * 2  ----->  i *= 2

lelouch@lelouch-IdeaPad-3-15IML05-D:~/Python/Workshops/python$
```

S.H.E.L.L.
SECURITY IS AN ILLUSION

Input

Open | input.py ~/Python/Workshops/python | Save

```
#! /usr/bin/python3

#In order to give input as user,we use input() function.

name = input('Enter Your Name : ')
print(name) #will print your input stored in the name variable

age = float(input('Enter Your age : '))
age = int(age)
print(age) #will print value stored in the age variable

#adding in a sentence
print("\n")
print("Printing with string concatination")
print("Welcome user, your name is "+name+" and your age is "+ str(age))
print("\n")
print("Printing with .format")
print("Welcome user, your name is {} and your age is {}".format(name,age))
print("\n")
print("Printing with f string")
print(f"Welcome user, your name is {name} and your age is {age}")
print("\n")
#Notice here we have to use str() for age and not for name,that's because age is a
integer variable and you cant add vartiables of different datatypes
print("Data-type of variable name and age")
print(type(name))
print(type(age))
print("\n-")
#we can change data type of an integer variable to string by using str(variable)
print("Changing data type of age from int to string")
print(type(str(age)))
```

Python 3 | Tab Width: 8 | Ln 1, Col 1 | INS

Input output Statements

```
lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Workshops/python
lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Works...
lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Works...
lelouch@lelouch-IdeaPad-3-15IML05-D:~/Python/Workshops/python$ python3 input.py
Enter Your Name : Rohan
Rohan
Enter Your age : 20
20


Printing with string concatination
Welcome user, your name is Rohan  and your age is 20


Printing with .format
Welcome user, your name is Rohan  and your age is 20


Printing with f string
Welcome user, your name is Rohan  and your age is 20


Data-type of variable name and age
<class 'str'>
<class 'int'>

-
Changing data type of age from int to string
<class 'str'>
lelouch@lelouch-IdeaPad-3-15IML05-D:~/Python/Workshops/python$
```

If else

conditions_if_else_elif.py
~/Python/Workshops/python

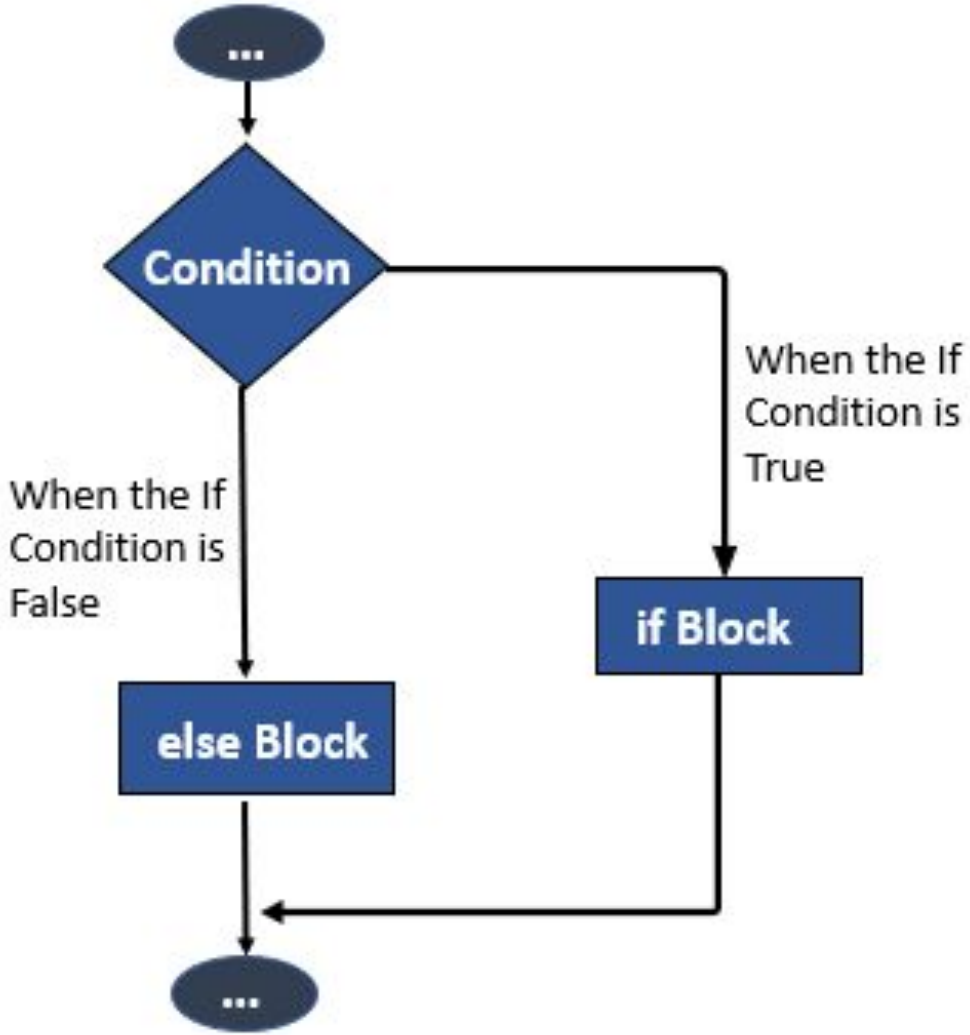
```
"""
>       -       greater than
<       -       less than
!=  -   not equal to
==    -  equal to
>=    -  greater than or equal to
<=      -       less than or equal to
"""


"""
"if" statement checks for statement on RHS and LHS and as an output is boolean data which is True or False.

if value returned by if statement is True then block of code inside "if" will execute, in same way canditions in elif are checked and at last if all condition fails else block is executed
"""

x = 1
y = 2

if x > y:
        print("x is greater than y")
        print("Inside if")
elif x == y:
        print("x is equal to y")
        print("Inside elif")
else:
        print("x is less than y")
        print("Inside else")
```


...
Condition
When the If Condition is True
When the If Condition is False
if Block
else Block
...

For loop

Open | for_loop.py ~/Python/Workshops/python | Save

```
#!/bin/python3

# for loop is a way to repeat process for known number of times
# last value ie 10 here is ignored
# range is a function which limits you values

"""
syntax of for loop
for variable in range(start,end,step_size):
    code_here
"""

# For loop with step size 1
# from 0 to 10 (0 inclusive 10 exclusive)
for i in range(10):
    print(i)
print("\n")
# for loop with step size 1 ended

# from 5 to 10
for i in range(5, 10):
    print(i)


print("\n")

# for loop with step size 2
for i in range(10,2):
    print(i)

print("\n")
# for loop with step size 2 ended

```

Python 3 | Tab Width: 8 | Ln 1, Col 1 | INS

Output

lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Workshops/python

lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Works...

lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Works...

```
lelouch@lelouch-IdeaPad-3-15IML05-D:~/Python/Workshops/python$ python3 for_loop.py
0
1
2
3
4
5
6
7
8
9


5
6
7
8
9


lelouch@lelouch-IdeaPad-3-15IML05-D:~/Python/Workshops/python$
```

While loop

Open | while_loop.py ~/Python/Workshops/python | Save

```
#!/bin/python3

"""
while loop also run as for but difference that in for loop you know number of times you want to run the loop, where as in while loop we give condition till which it will run.
"""

"""
syntax of while loop

condition_variable
while condition:
    code_here
    condition_variable_modification
"""

# while loop that prints from 0 to 10
i = 0
while i < 10:
    print(i)
    i += 1
print("\n")

i = 5
while i < 10:
    print(i)
    i += 1
print("\n")



i=1
while i < 10:
    print(i)
    i += 2

print("\n")
```

While output

lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Workshops/python

lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Works...

lelouch@lelouch-IdeaPad-3-15IML05-D: ~/Python/Works...

```
lelouch@lelouch-IdeaPad-3-15IML05-D:~/Python/Workshops/python$ python3 while_loop.py
0
1
2
3
4
5
6
7
8
9


5
6
7
8
9


1
3
5
7
9

lelouch@lelouch-IdeaPad-3-15IML05-D:~/Python/Workshops/python$
```

List

Open | list.py ~/Python/Workshops/python | Save

```
#! /usr/bin/python3

#list are used to stored multiple items in a single variable

a = ["1","2","3","4","5","6"] #square brackets are used in list

#printing a list
print(a)
print("\n") #used to add a new line

#printing a value at a certain index,in list index start froms 0
print(a[0]) #prints first item in list
print(a[1]) #prints second item in list
print(a[2]) #prints third item in list
print(a[3]) #prints fourth item in list
print(a[4]) #prints fifth item in list
print(a[5]) #prints sixth item in list
print(a[-1]) # Negative integers can be used to print from right to left,will print first word from right.
print("\n")

#printing a part of list, like first 2 or first 3 items
print(a[0:2]) #prints the first two items in the list.
print(a[0:4]) #prints the first four items in the list.
print("\n")

#adding items in list using append and insert

a.append("7") #will add Lamborghini item in list after the last item in list
print(a)
a.insert(0,'0') #with insert,you can specify the index too,0 will be the first here now .
print(a)
print("\n")

#removing items from list
a.pop(5) #will remove the item '5',if it exist in the list.
print(a)
a.remove("7") #will remove the item '7',if it exist in the list.
print(a)
```

Output

```
lelouch@lelouch-IdeaPad-3-15IML05-D:~/Python/Workshops/python$ python3 list.py
['1', '2', '3', '4', '5', '6']


1
2
3
4
5
6
6


['1', '2']
['1', '2', '3', '4']


['1', '2', '3', '4', '5', '6', '7']
['0', '1', '2', '3', '4', '5', '6', '7']


['0', '1', '2', '3', '4', '6', '7']
['0', '1', '2', '3', '4', '6']
lelouch@lelouch-IdeaPad-3-15IML05-D:~/Python/Workshops/python$
```

What Next

- Try-except , Recursion
- Keylogger