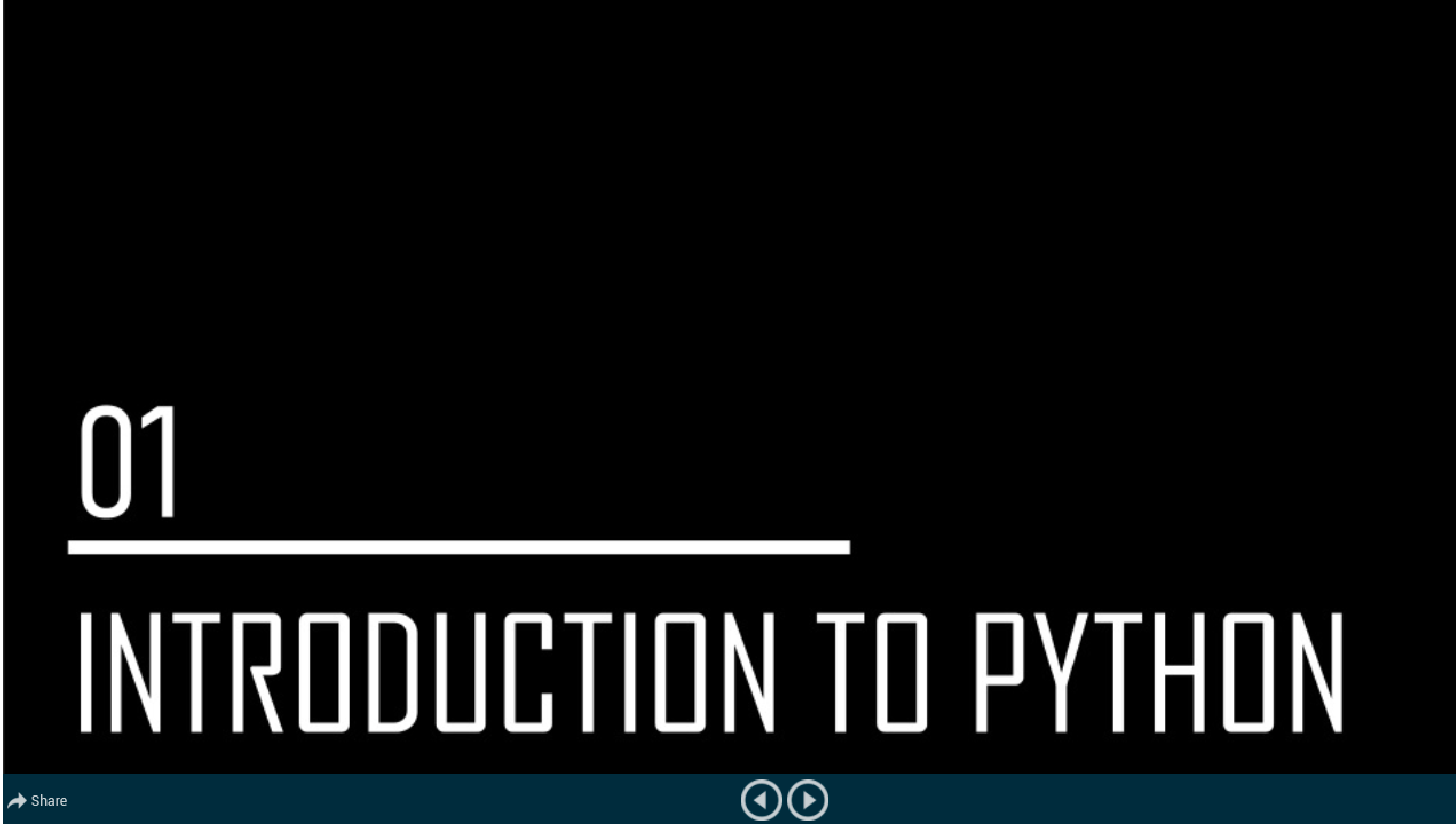


01

INTRODUCTION TO PYTHON

PYTHON PROGRAMMING

OPEN-SOURCE

TONS OF LIBRARIES

PROCEDURE ORIENTED

OBJECT-ORIENTED

INTERPRETED

EASY TO LEARN

COMMENTS IN PYTHON

Print('Hello world')

this is a single line comment

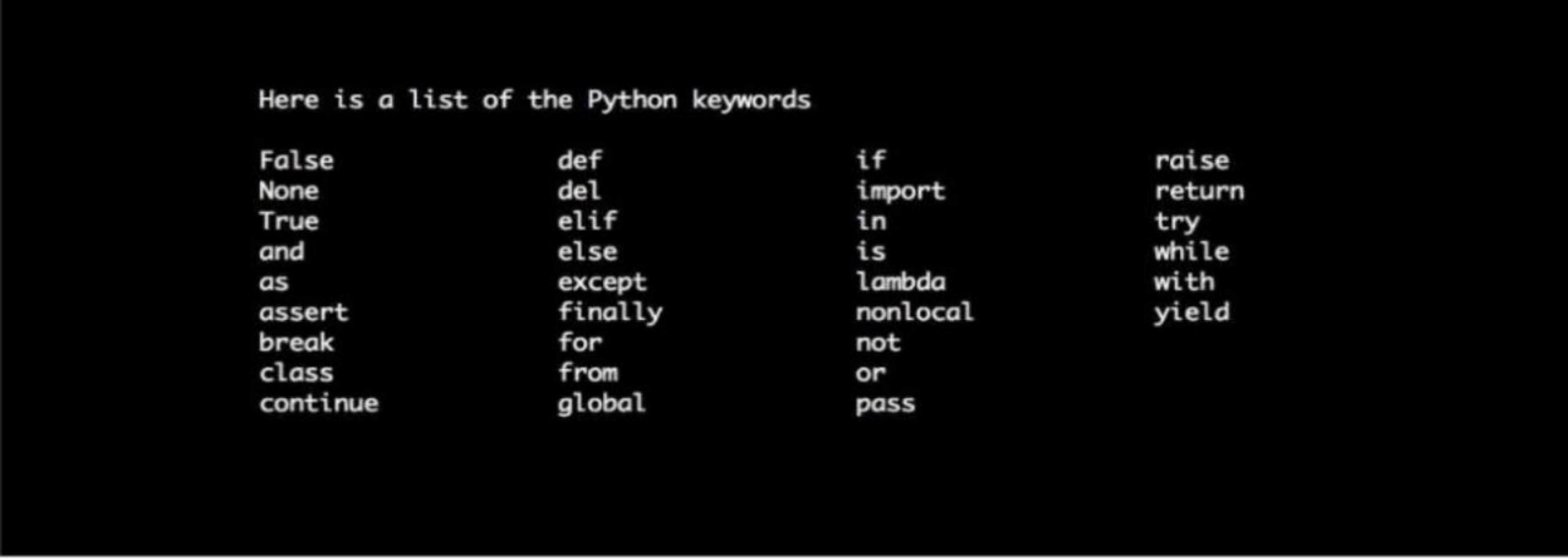
...
this is a multi-line comme
The difference is uncann
...

Comments in python

There are two types of comments in python

1. Single line comment
2. Multi line comment

Share

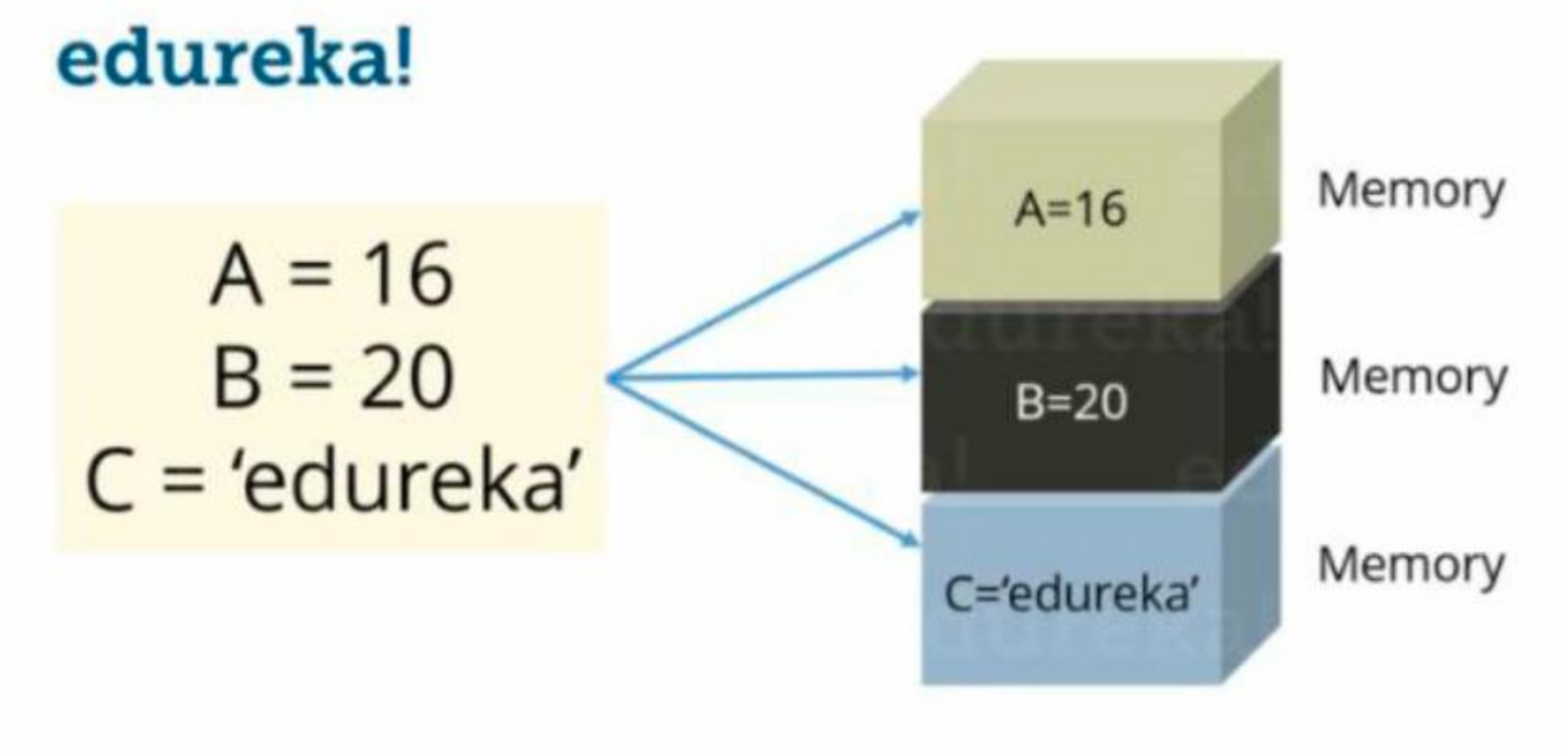


```
Here is a list of the Python keywords

False       def         if          raise
None        del         import      return
True        elif        in          try
and         else        is          while
as          except      lambda      with
assert      finally     nonlocal    yield
break       for         not
class       from        or
continue    global      pass
```

Keyword and Identifiers

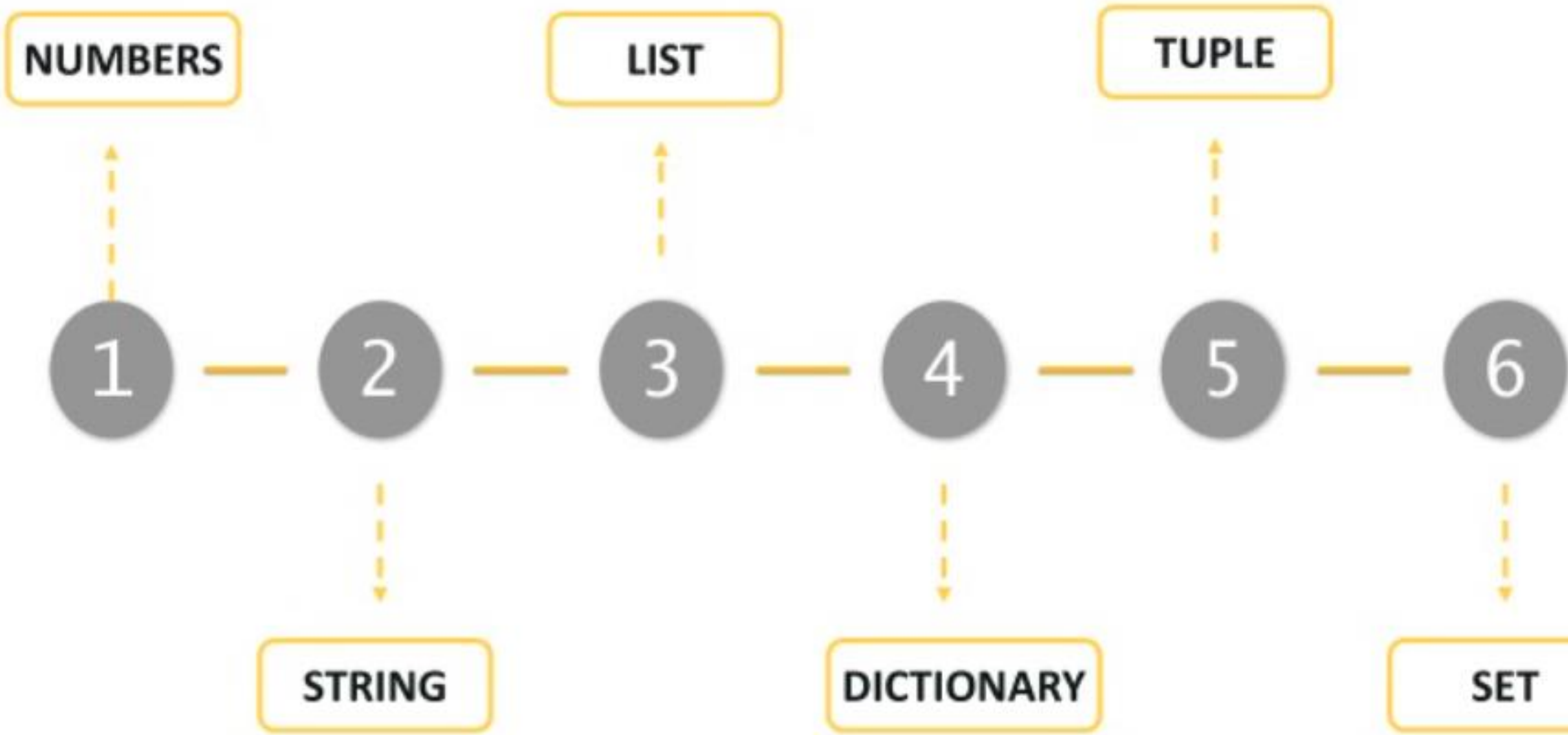
Keywords in python are special words that have a specific meaning. And identifiers are user defined names to represen
a variable, function, a class or a module.

WHAT IS A VARIABLE?



```
A = 16
B = 20
C = 'edureka'
```

DATA TYPES IN PYTHON



1 NUMBERS

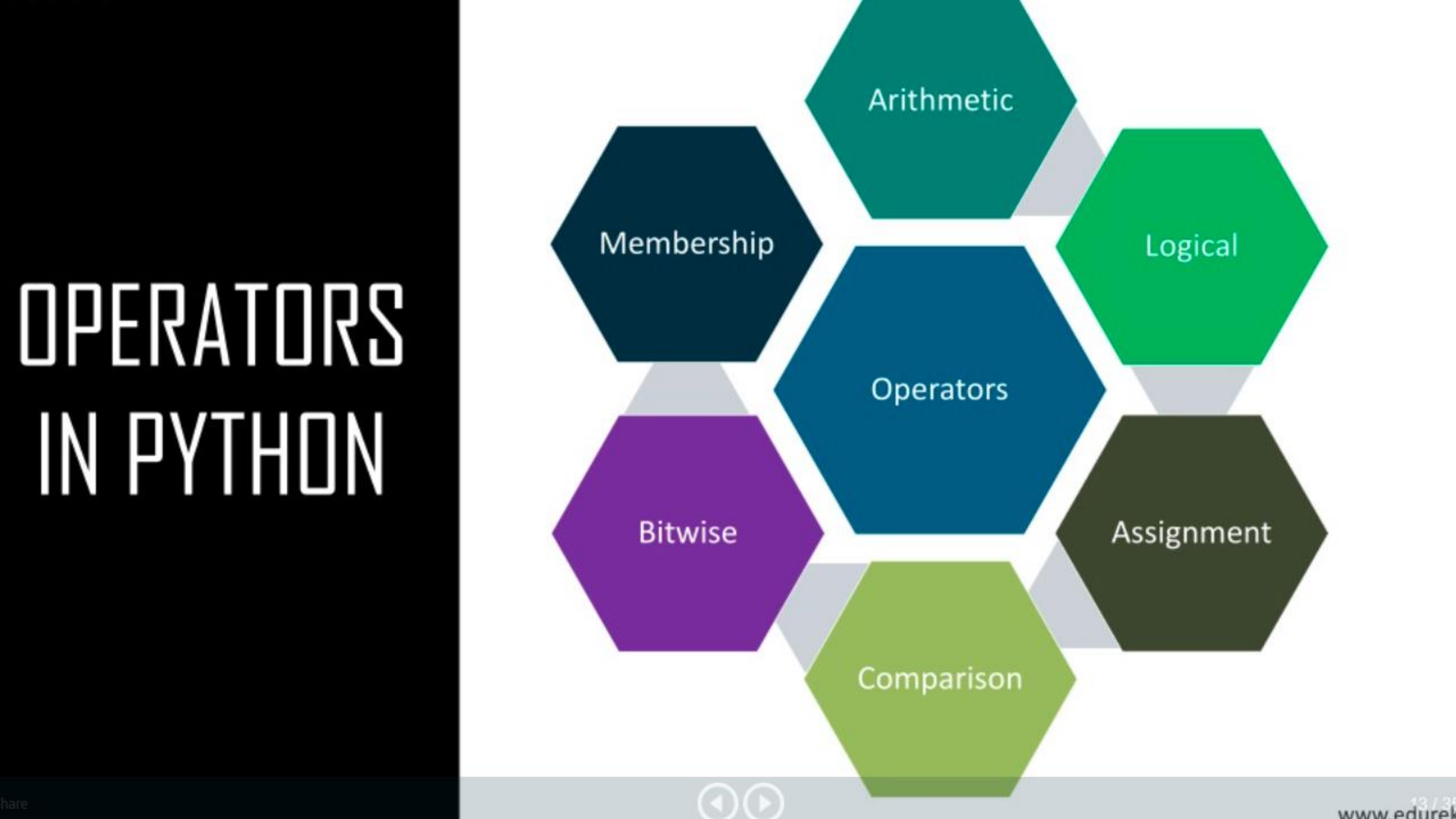
2 STRING

3 LIST

4 DICTIONARY

5 TUPLE

6 SET



OPERATORS IN PYTHON

Operators

Arithmetic

Logical

Assignment

Comparison

Bitwise

Membership

Why Should We Use Loops?

Suppose you want to print the numbers from 0-10

0,1,2,3,4,5,6,7,8,9,10

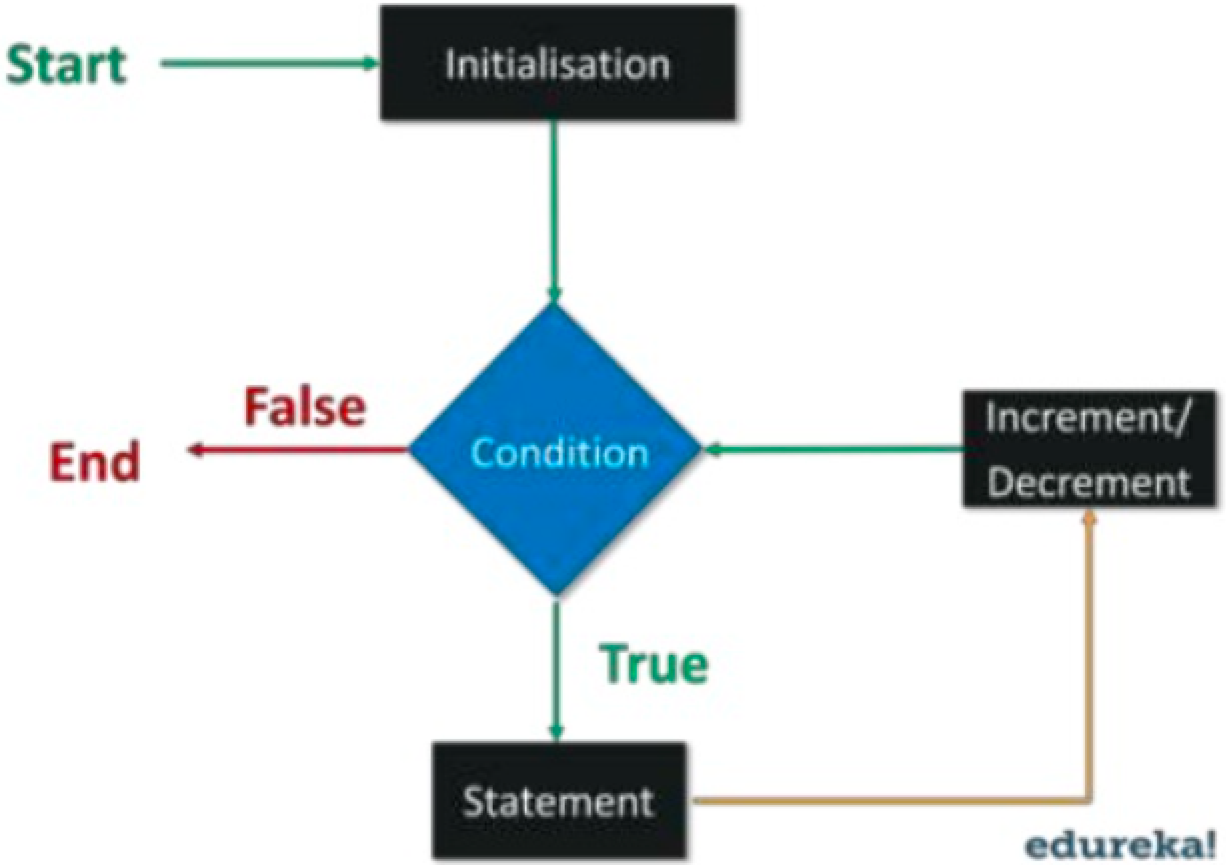
OR

You want to print the sum of all even numbers until 100.

OR

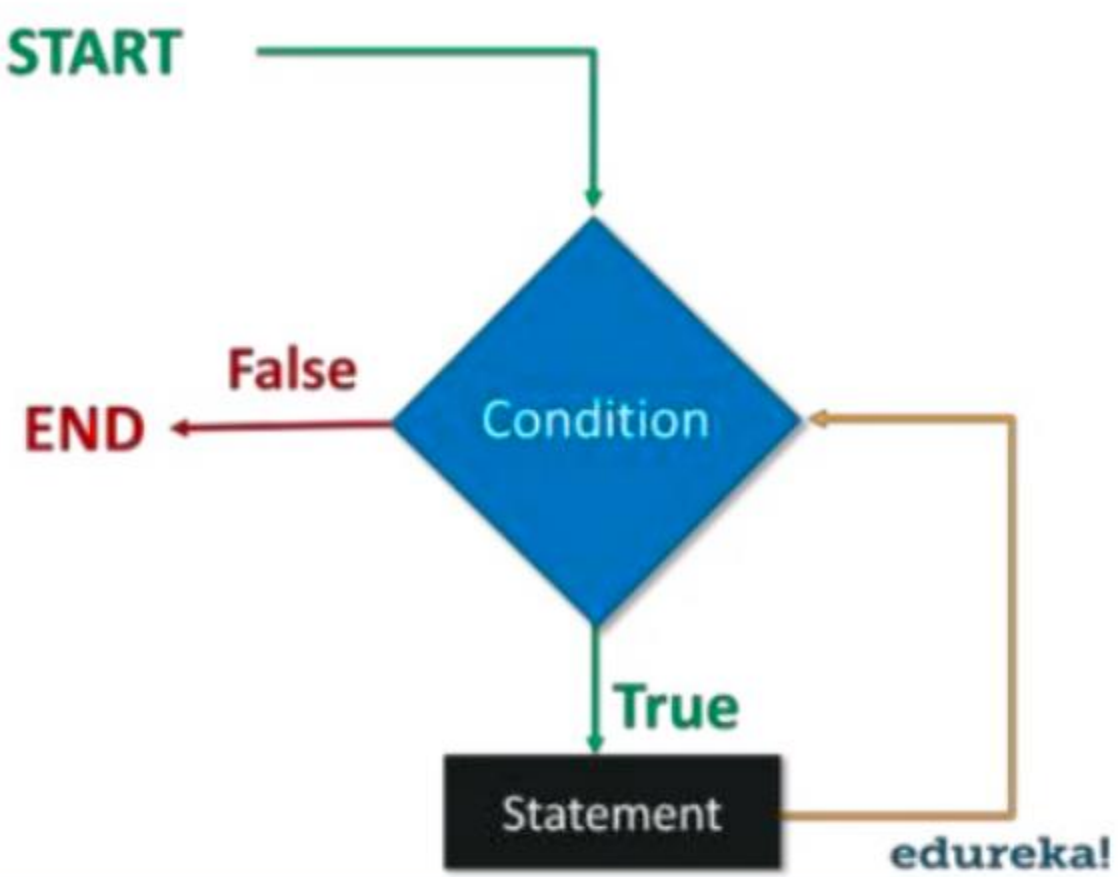
You want to print the name of all employees in a sequence until you come across lets say 'Ravi Mehta'

FOR LOOP



Why FOR Loop?
When we are iterating over a
sequence.
You know the number of
times the statements will get
executed.

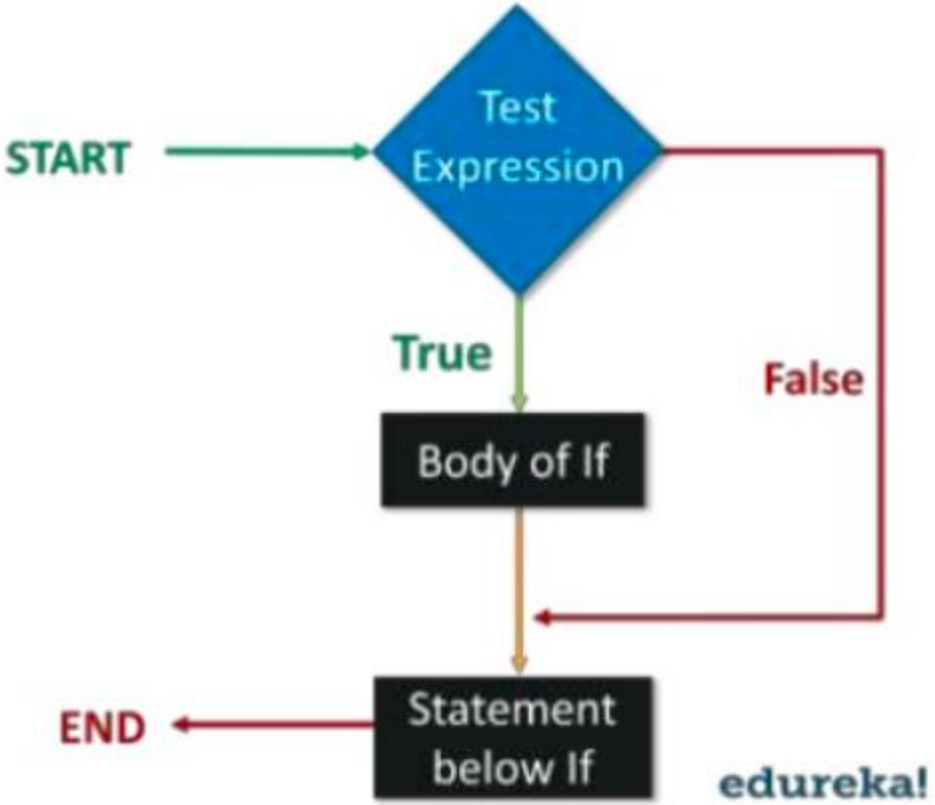
WHILE LOOP

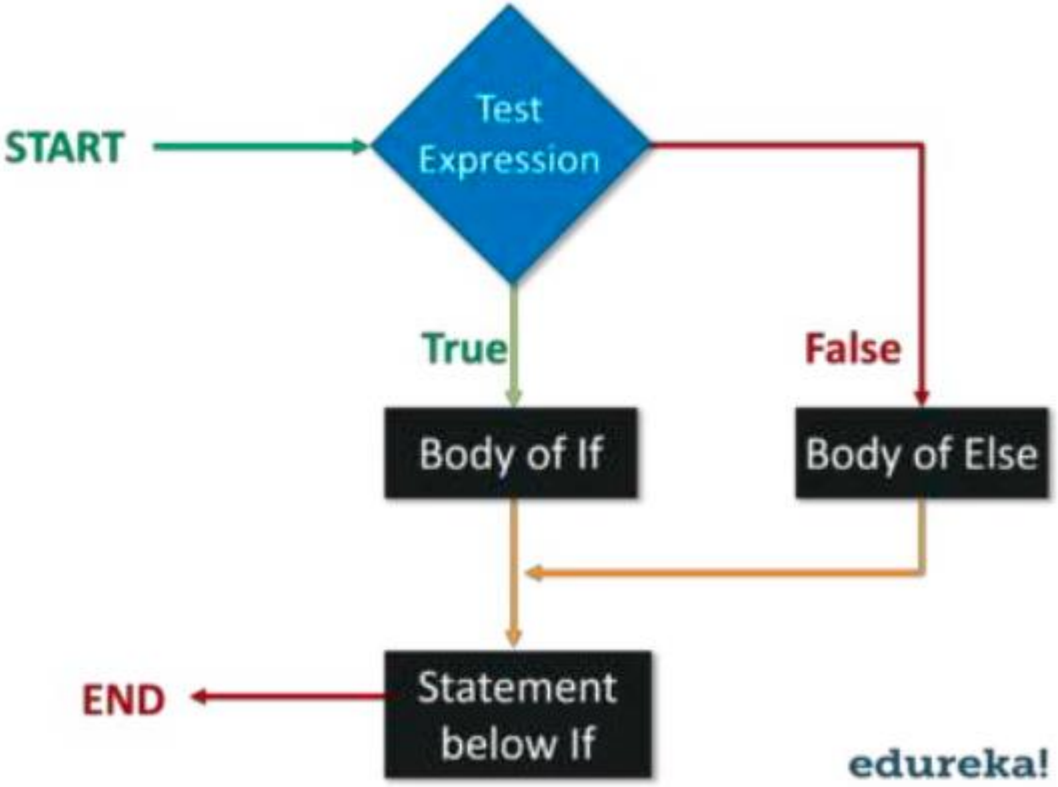


Why WHILE Loop?
When we are executing a set of statements , if the condition holds true.

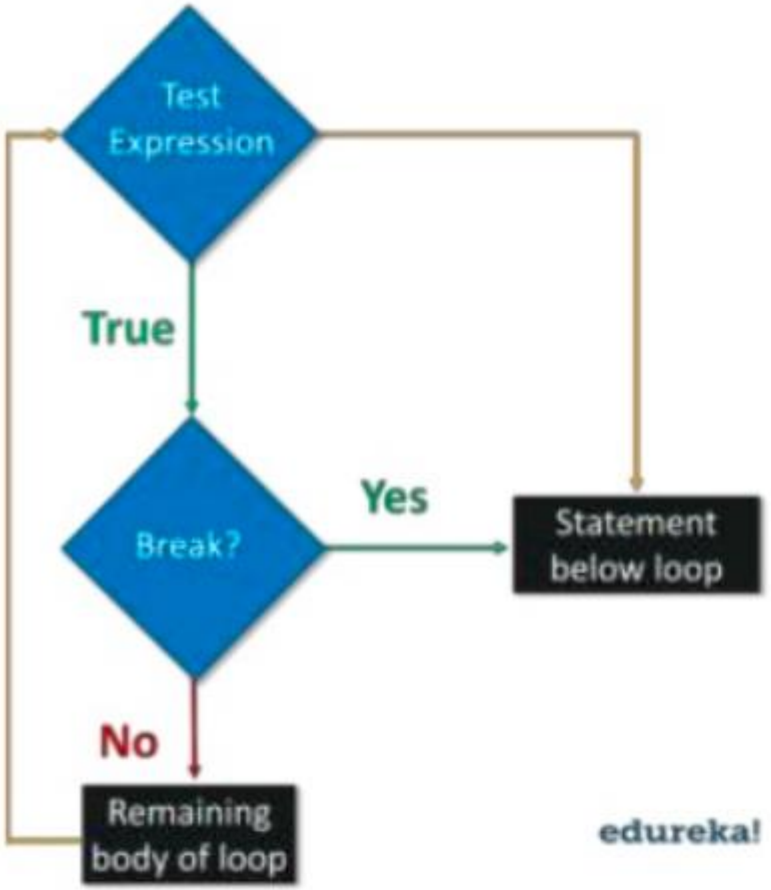
The execution stops as soon as the condition is false.

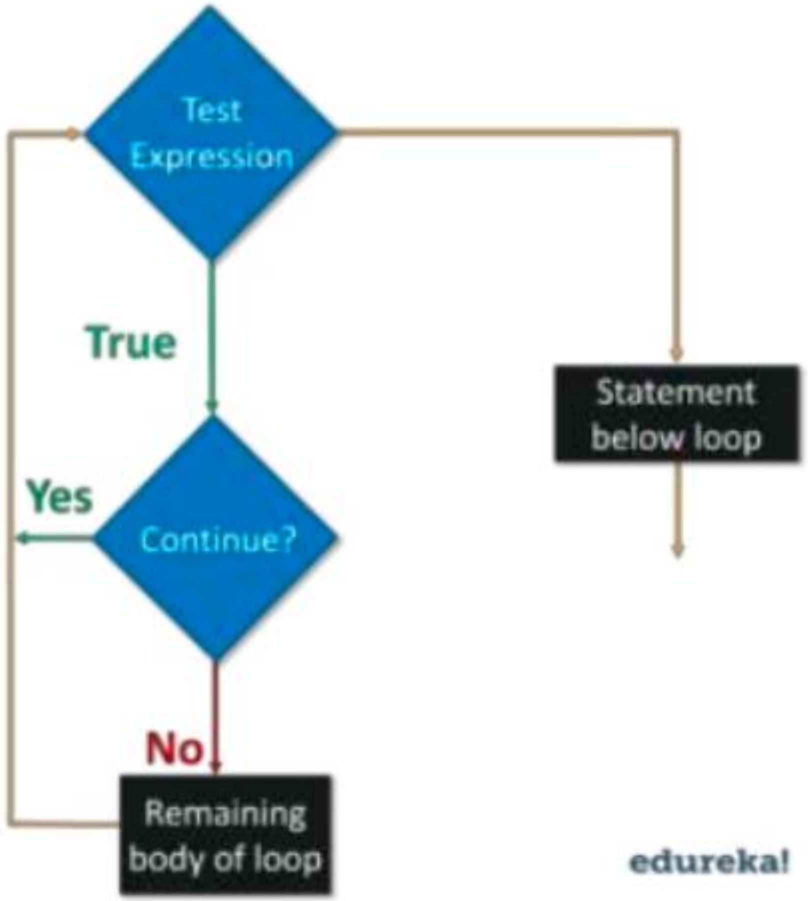
CONDITIONAL STATEMENTS



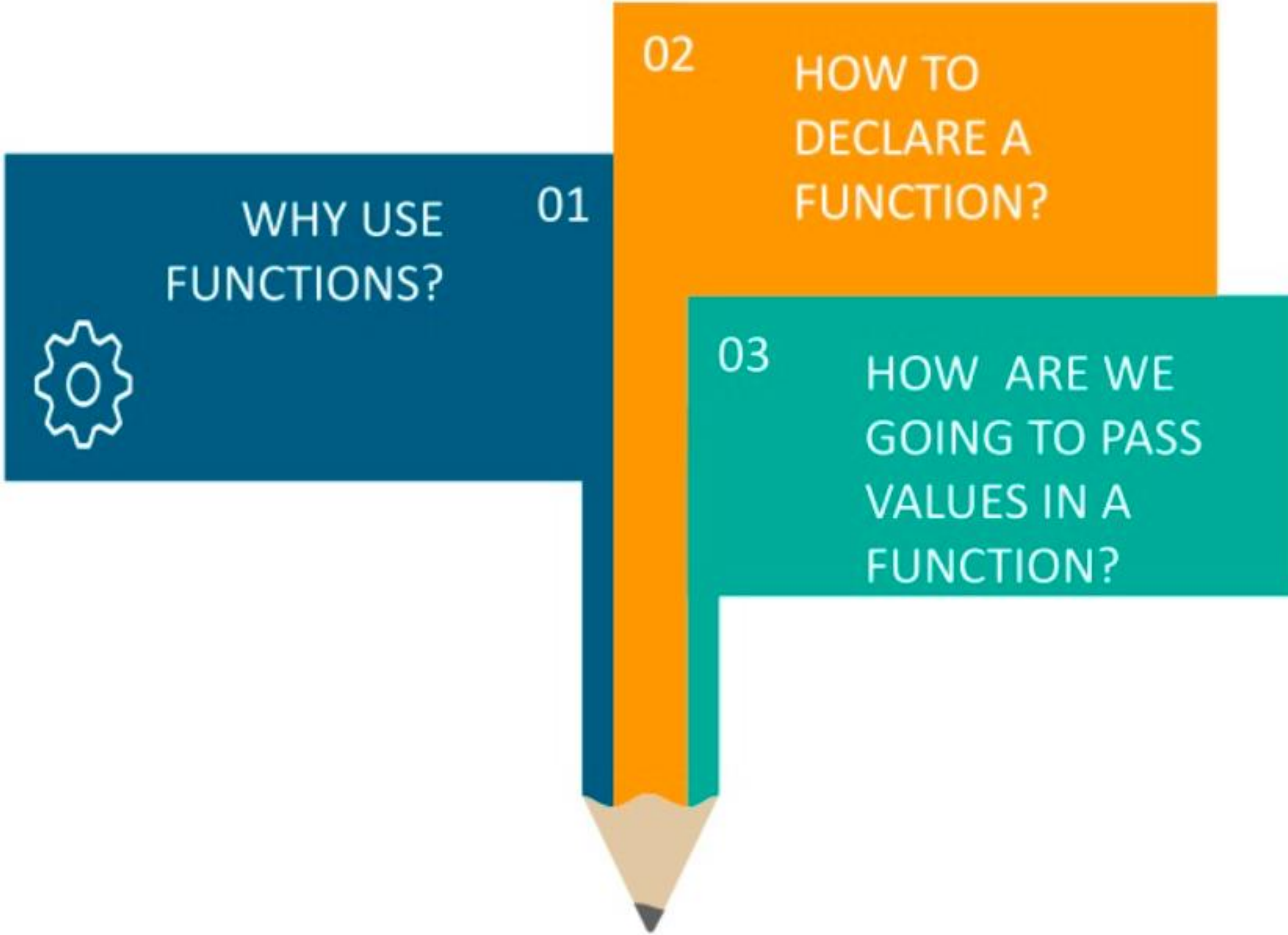


CONTROL STATEMENTS





FUNCTIONS IN PYTHON



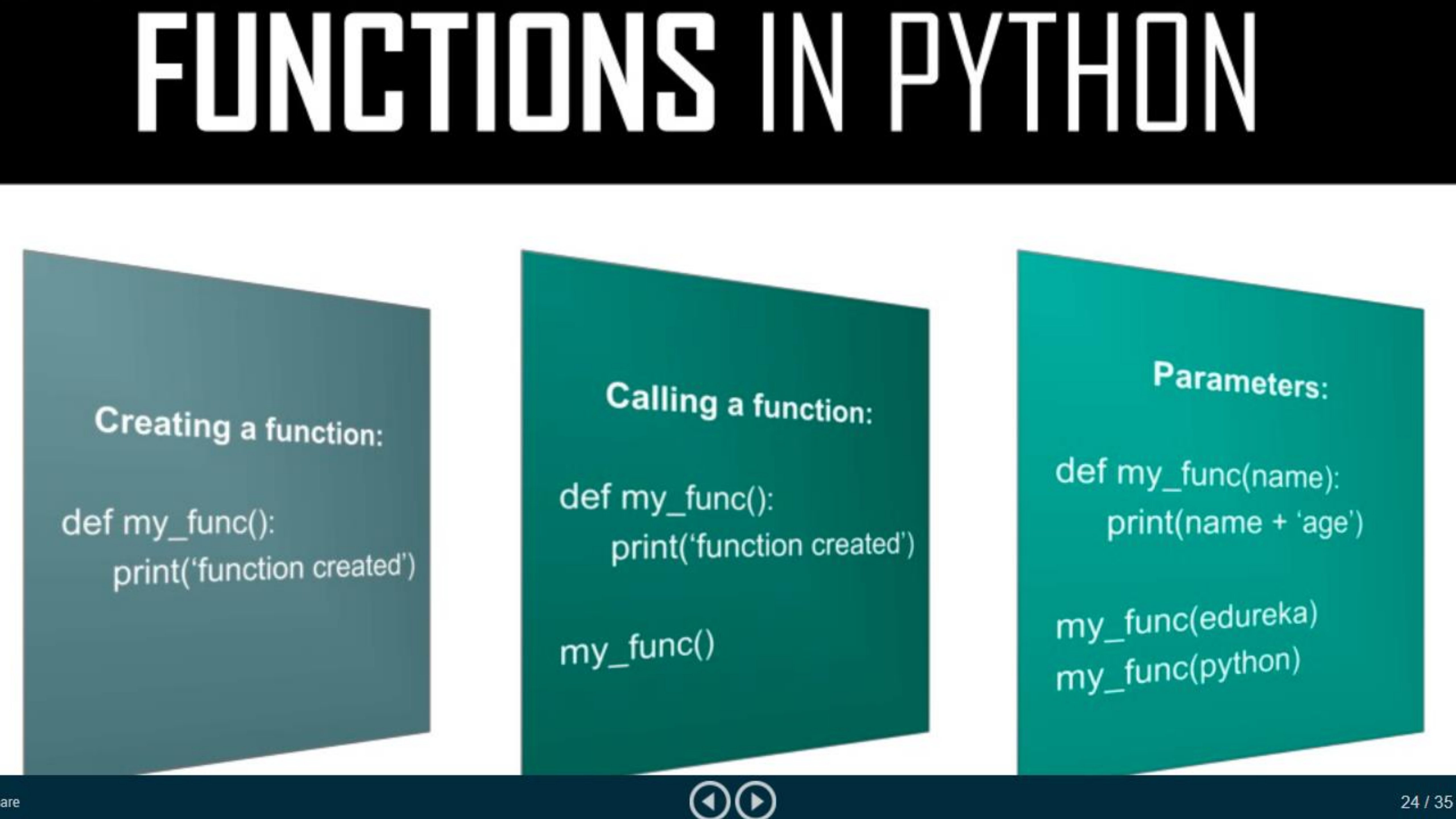
01 WHY USE FUNCTIONS?

02 HOW TO DECLARE A FUNCTION?

03 HOW ARE WE GOING TO PASS VALUES IN A FUNCTION?

Why Use Functions?

Instead You can define a function to calculate factorial. And you can call that function every time you need to calculate factorial.



FUNCTIONS IN PYTHON

Creating a function:

```
def my_func():
    print('function created')
```

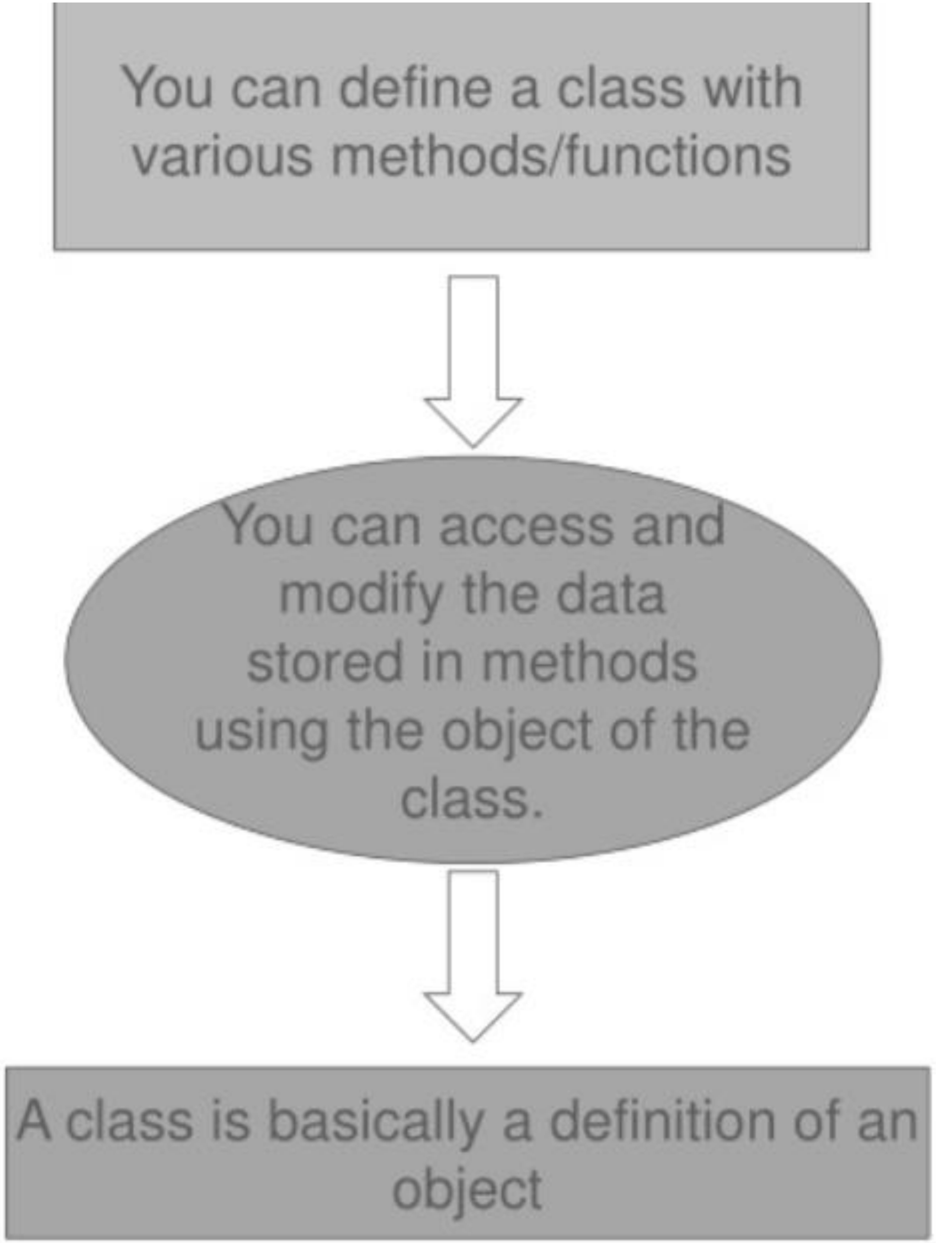
Calling a function:

```
def my_func():
    print('function created')

my_func()
```

Parameters:

```
def my_func(name):
    print(name + 'age')

my_func(edureka)
my_func(python)
```

Why Use Classes?



You can define a class with various methods/functions

You can access and modify the data stored in methods using the object of the class.

A class is basically a definition of an object

CLASSES & OBJECTS

Creating a class:

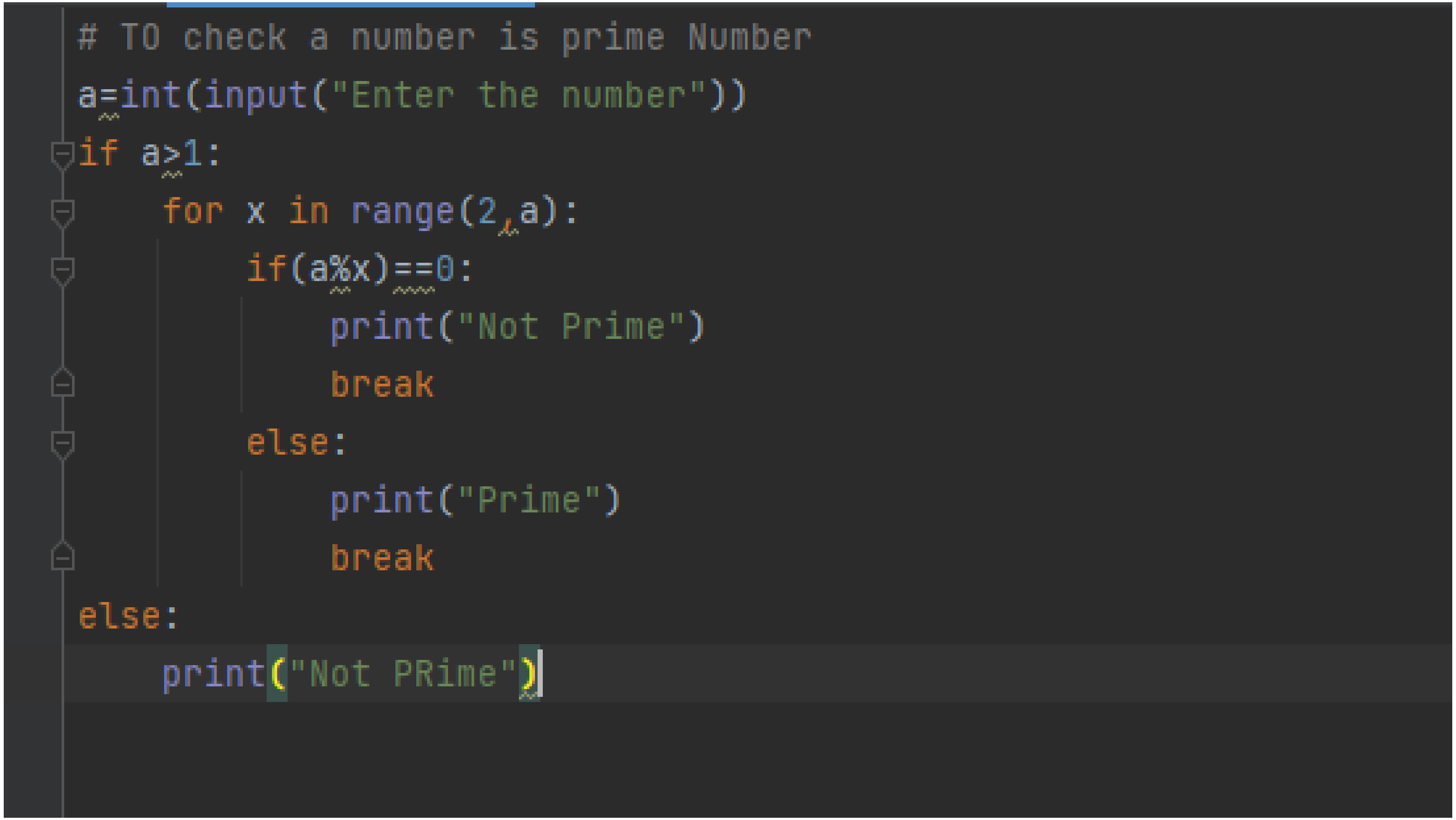
```
class Myclass:
    name =‘edureka’
```

Creating an object:

```
ob1 = Myclass()
print(ob1.name)
```

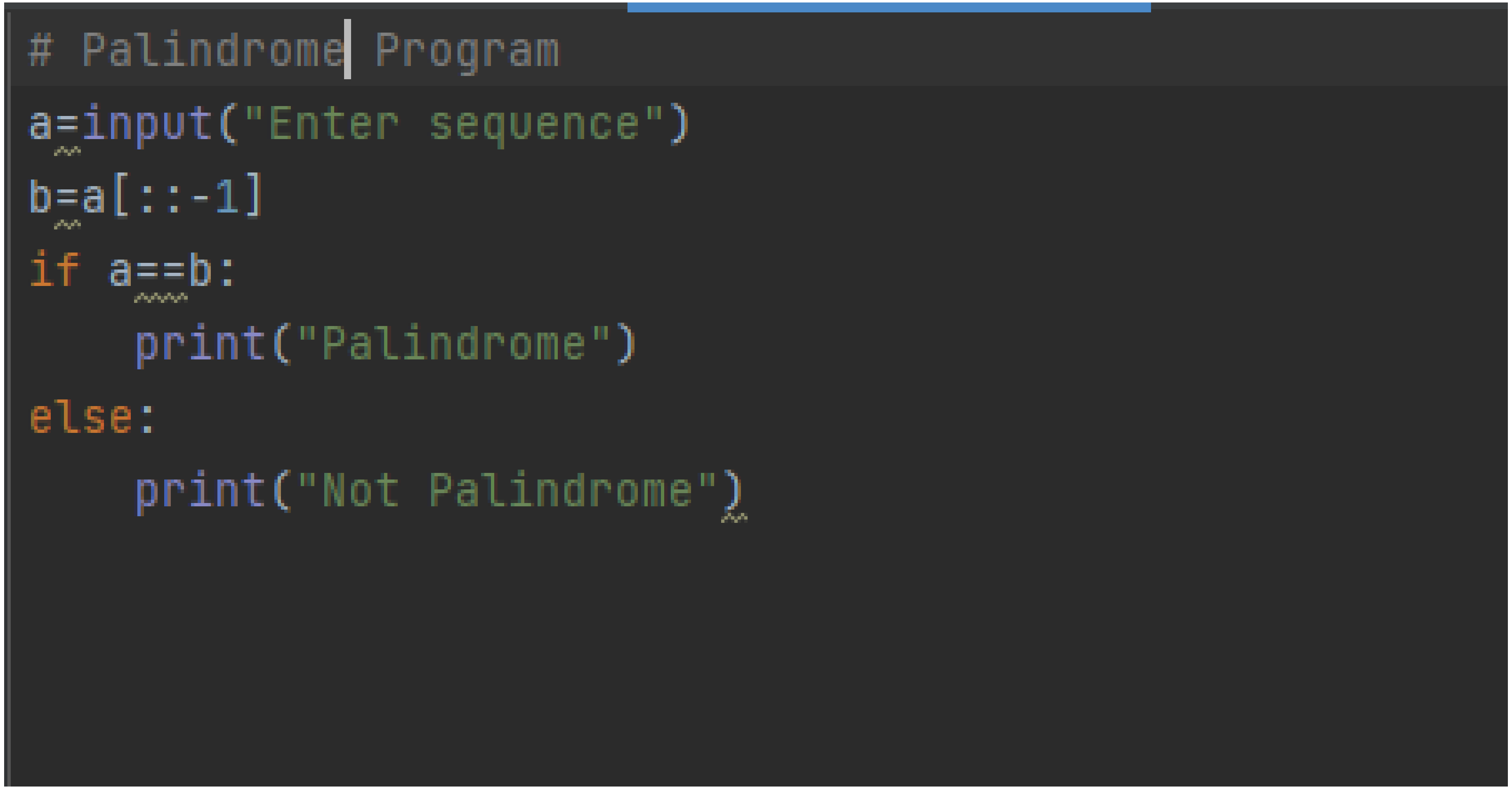
__init__()

Prime Number Program



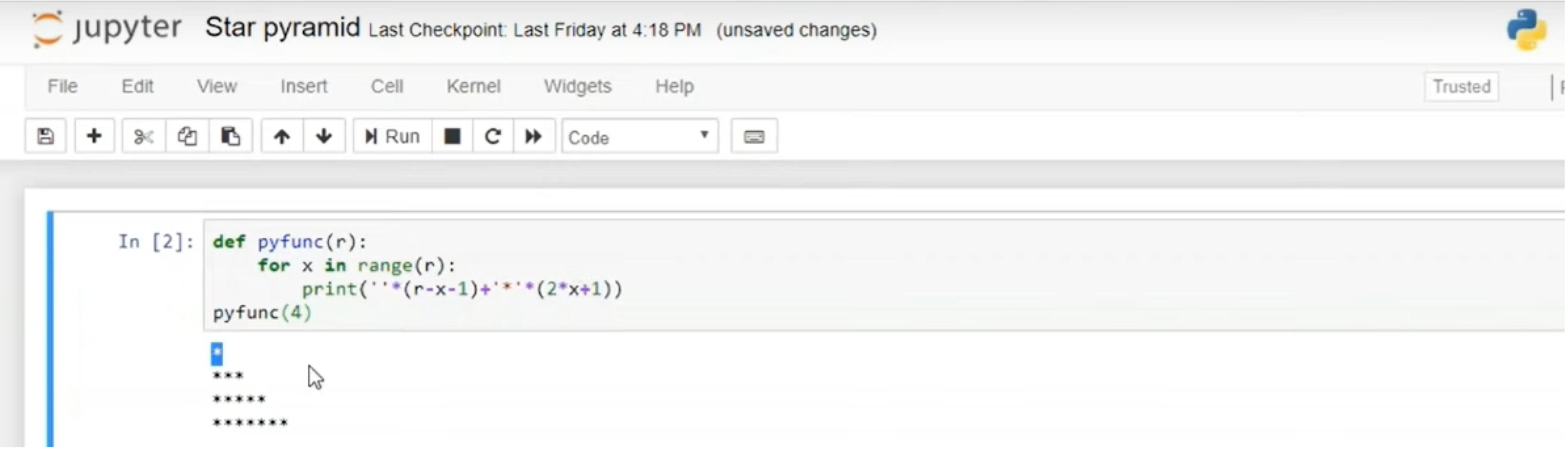
```
# TO check a number is prime Number
a=int(input("Enter the number"))
if a>1:
    for x in range(2,a):
        if(a%x)==0:
            print("Not Prime")
            break
        else:
            print("Prime")
            break
else:
    print("Not PRime")
```

Palindrome Program



```
# Palindrome Program
a=input("Enter sequence")
b=a[::-1]
if a==b:
    print("Palindrome")
else:
    print("Not Palindrome")
```

Pyramid Program

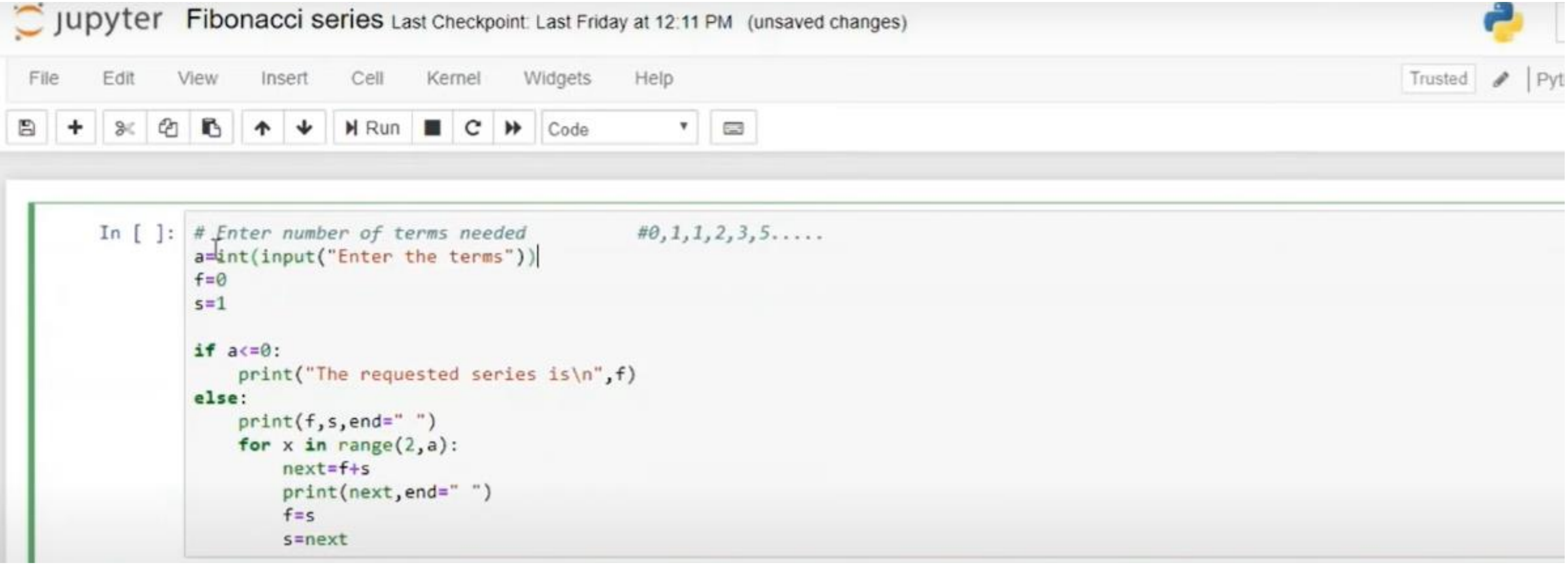

jupyter
Star pyramid Last Checkpoint: Last Friday at 4:18 PM (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Run
Code

In [2]:

```
def pyfunc(r):
    for x in range(r):
        print(''*(r-x-1)+'*'*(2*x+1))
pyfunc(4)
```

```
*
***
*****
*******
```

Fibonacci Series


jupyter
Fibonacci series Last Checkpoint: Last Friday at 12:11 PM (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Pyt
Run
Code
In [ ]:

```
# Enter number of terms needed          #0,1,1,2,3,5.....
a=int(input("Enter the terms"))
f=0
s=1

if a<=0:
    print("The requested series is\n",f)
else:
    print(f,s,end=" ")
    for x in range(2,a):
        next=f+s
        print(next,end=" ")
        f=s
        s=next
```