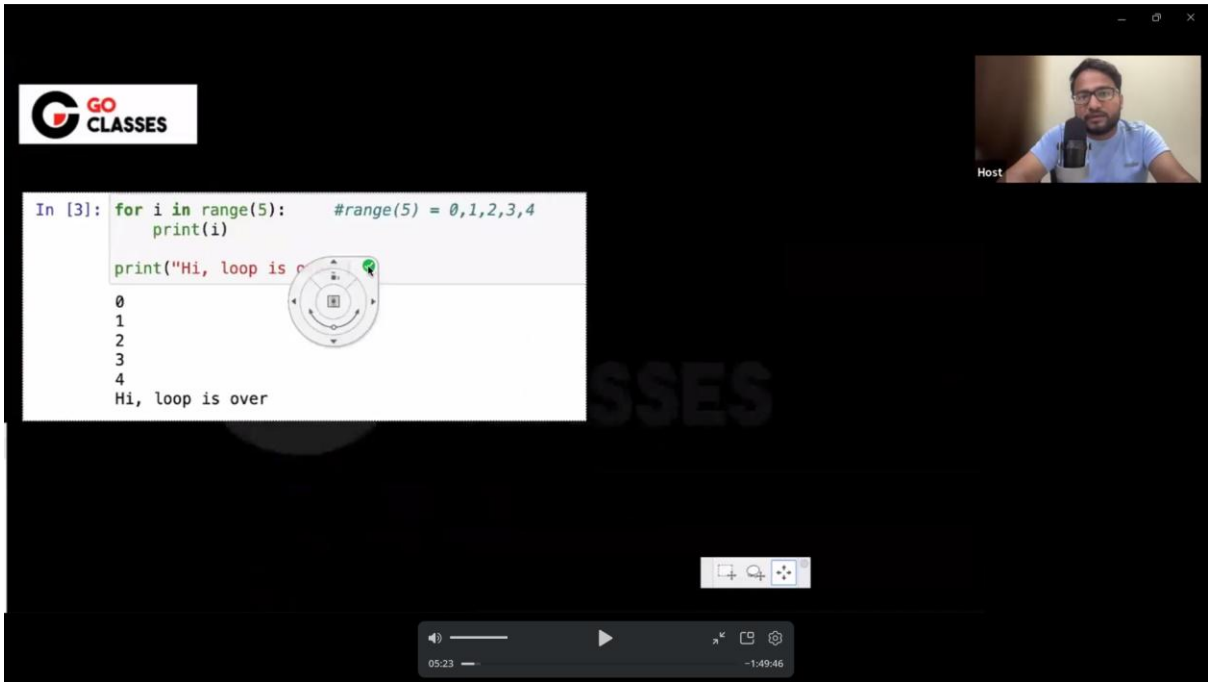
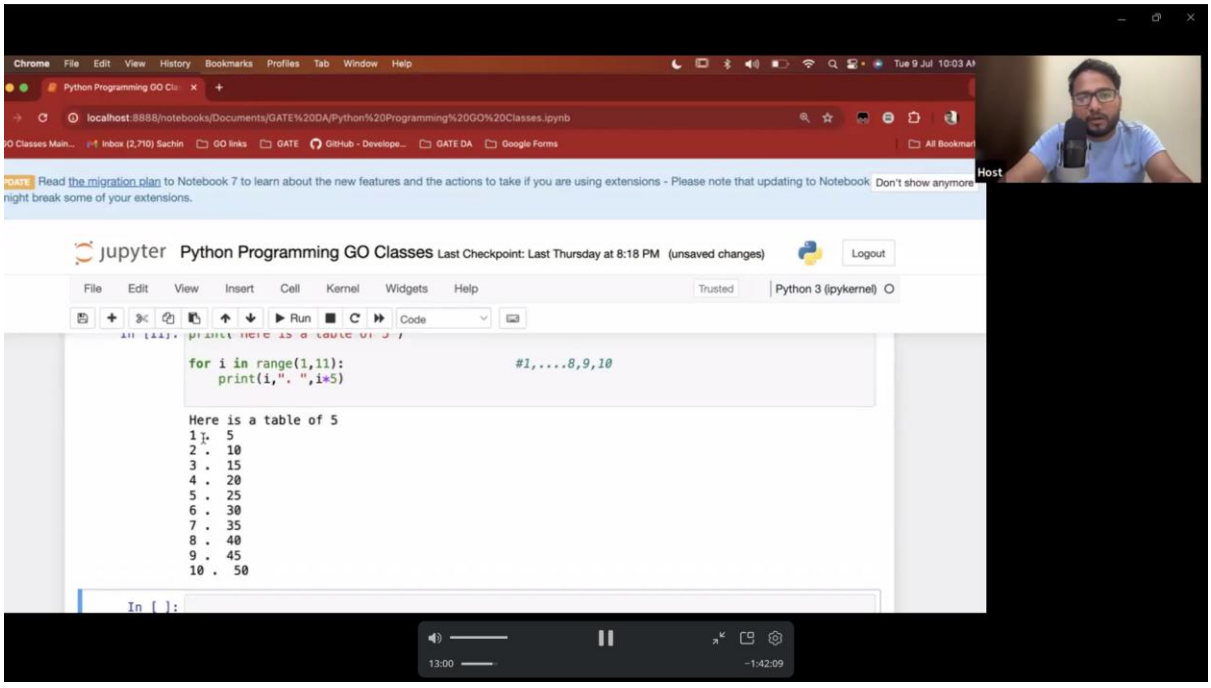
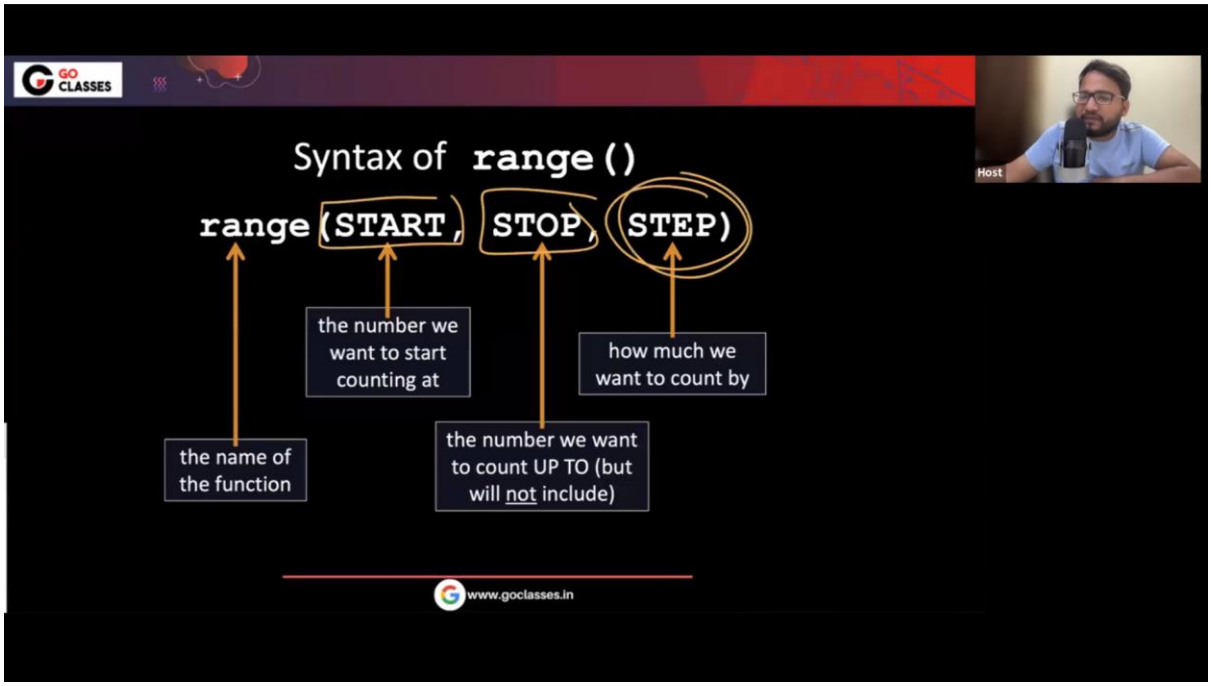
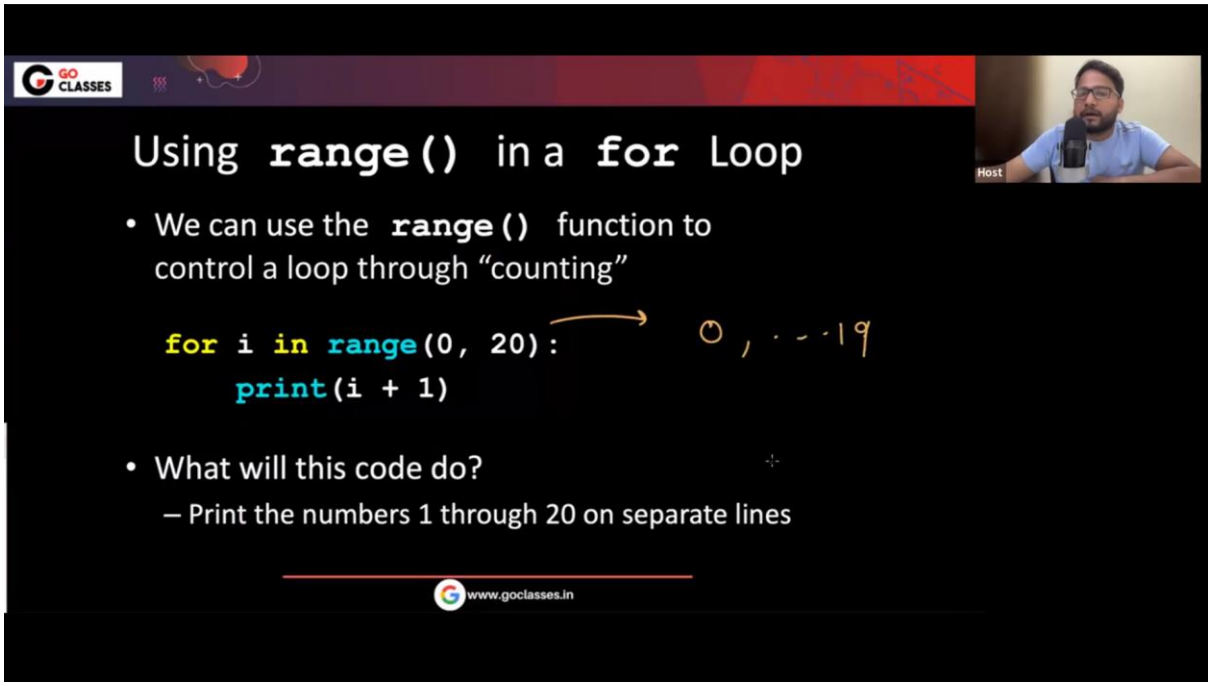
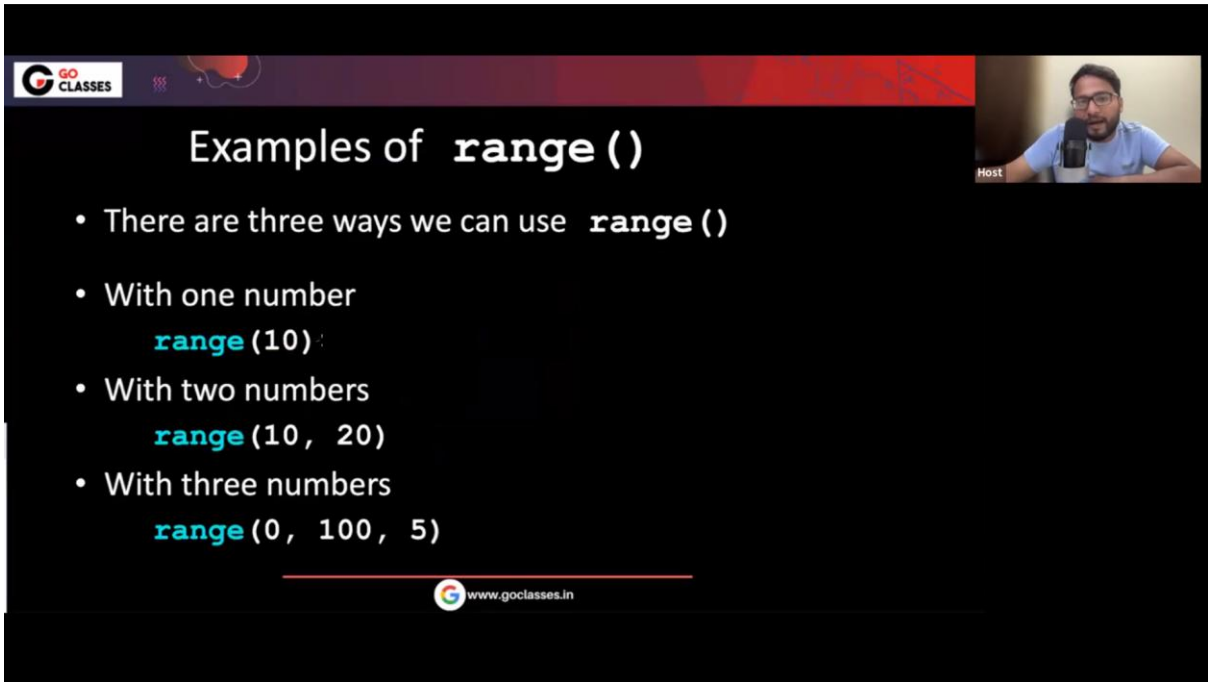
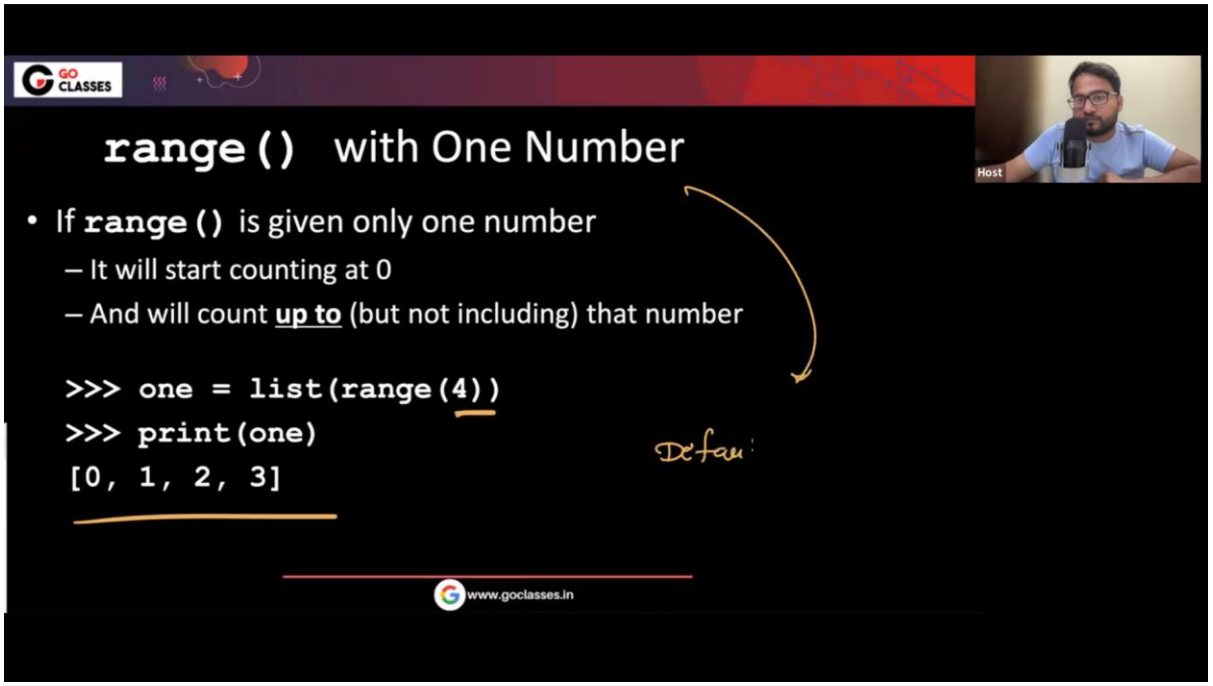

GO CLASSES
Host
In [3]:
for i in range(5): #range(5) = 0,1,2,3,4
print(i)
print("Hi, loop is
0
1
2
3
4
Hi, loop is over
05:23
-1:49:46

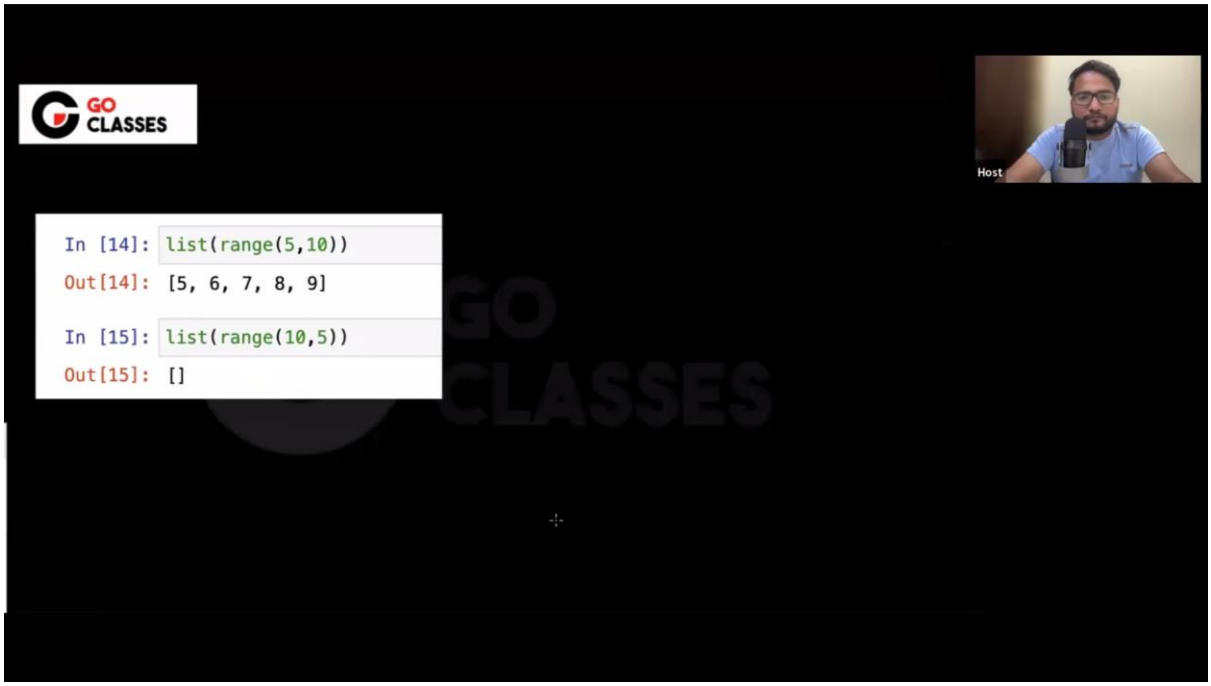

Host
Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook might break some of your extensions.
Don't show anymore
jupyter Python Programming GO Classes Last Checkpoint: Last Thursday at 8:18 PM (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
for i in range(1,11): #1,....8,9,10
print(i,". ",i*5)
Here is a table of 5
1 . 5
2 . 10
3 . 15
4 . 20
5 . 25
6 . 30
7 . 35
8 . 40
9 . 45
10 . 50
In [ ]:
13:00
-1:42:09

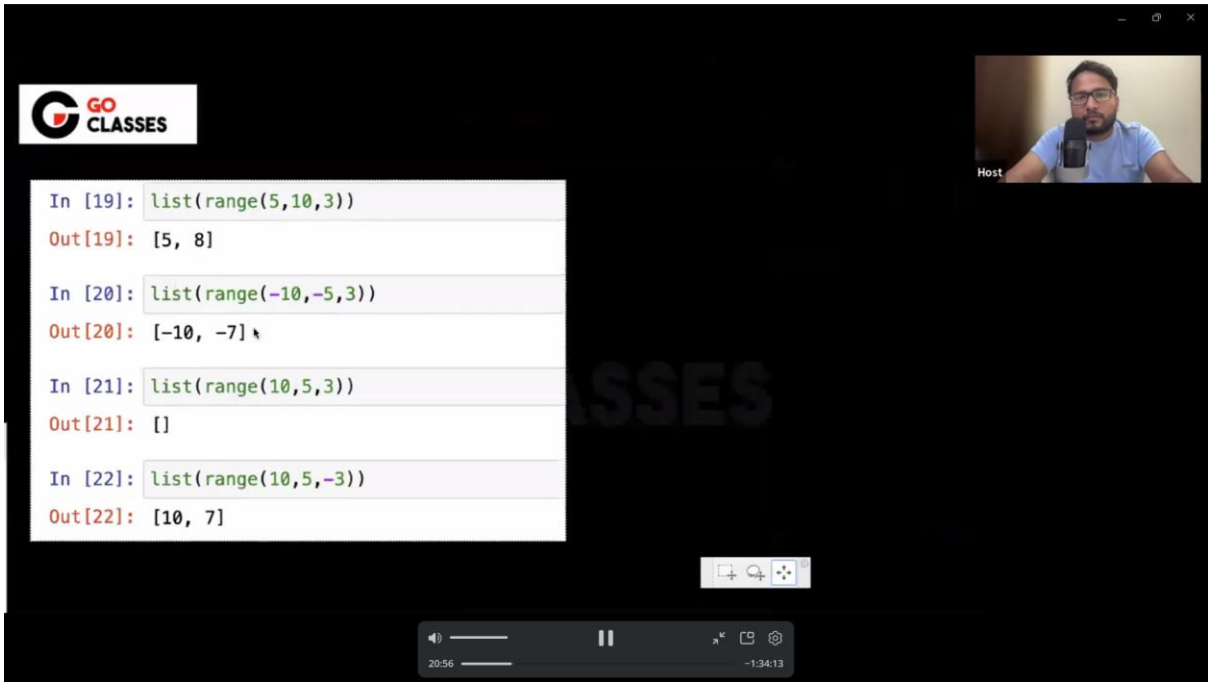

GO CLASSES
Host
Syntax of range()
range(START, STOP, STEP)
the number we want to start counting at
how much we want to count by
the name of the function
the number we want to count UP TO (but will not include)
www.goclasses.in

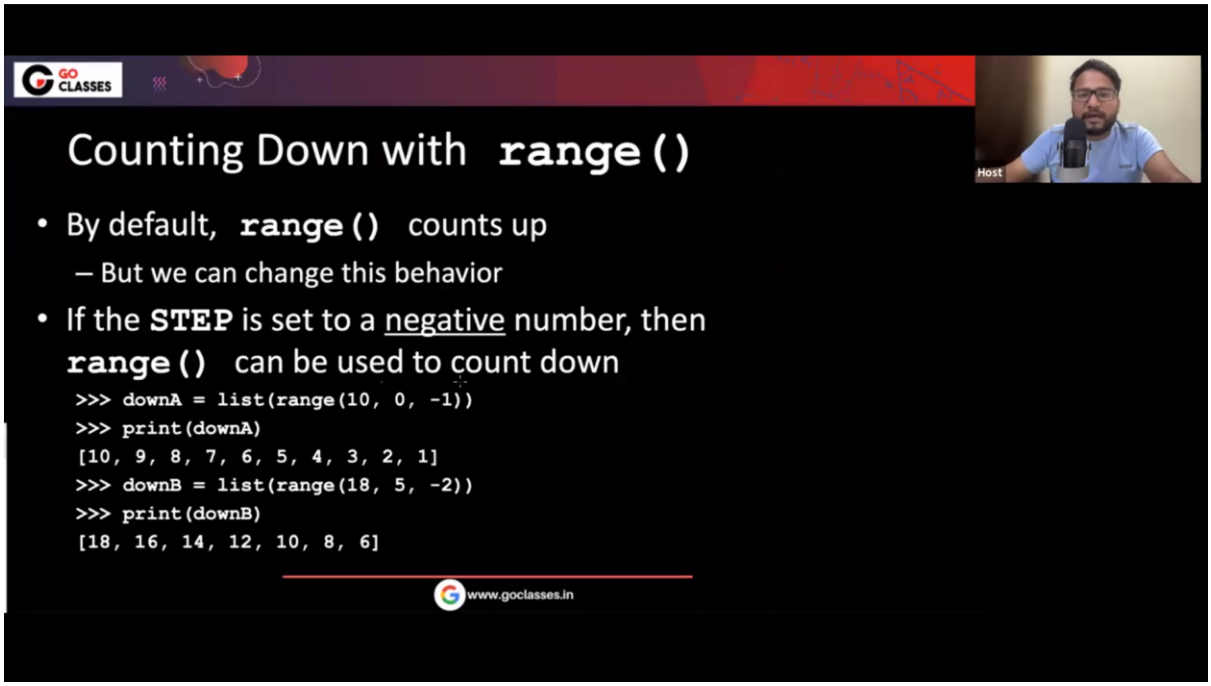

GO CLASSES
Host
Using range() in a for Loop
• We can use the range() function to control a loop through "counting"
for i in range(0, 20):
print(i + 1)
0, ..., 19
• What will this code do?
– Print the numbers 1 through 20 on separate lines
www.goclasses.in

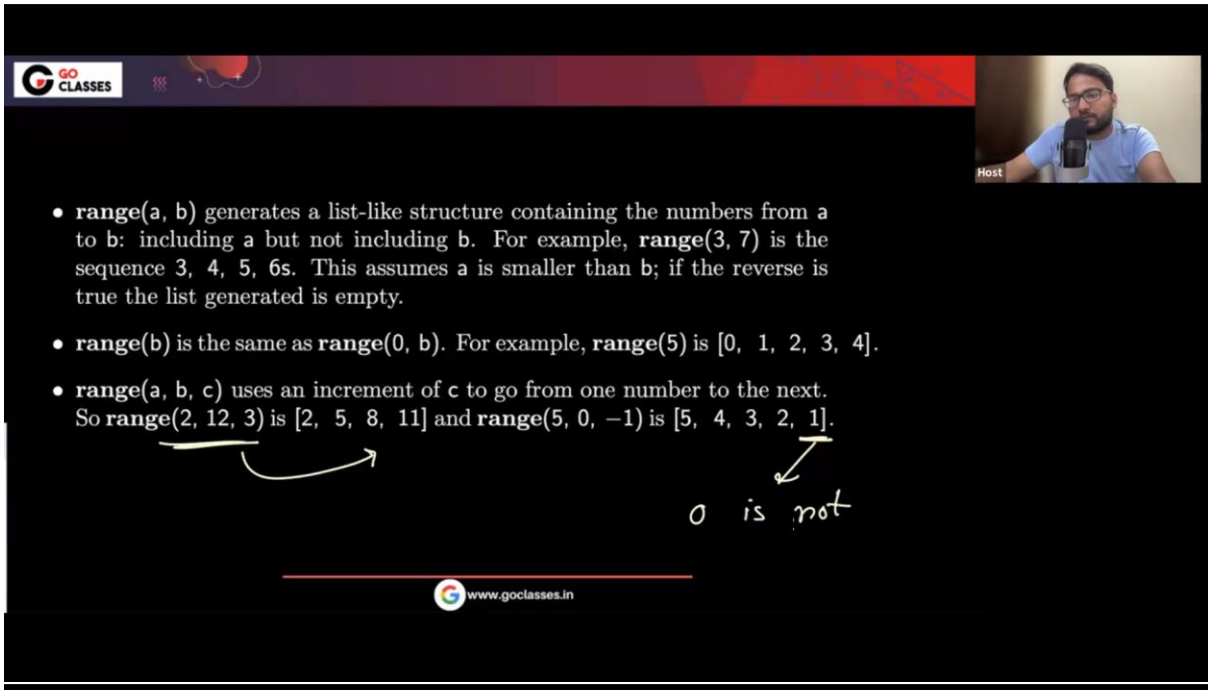

GO CLASSES
Examples of range()
• There are three ways we can use range()
• With one number
range(10)
• With two numbers
range(10, 20)
• With three numbers
range(0, 100, 5)
www.goclasses.in
Host

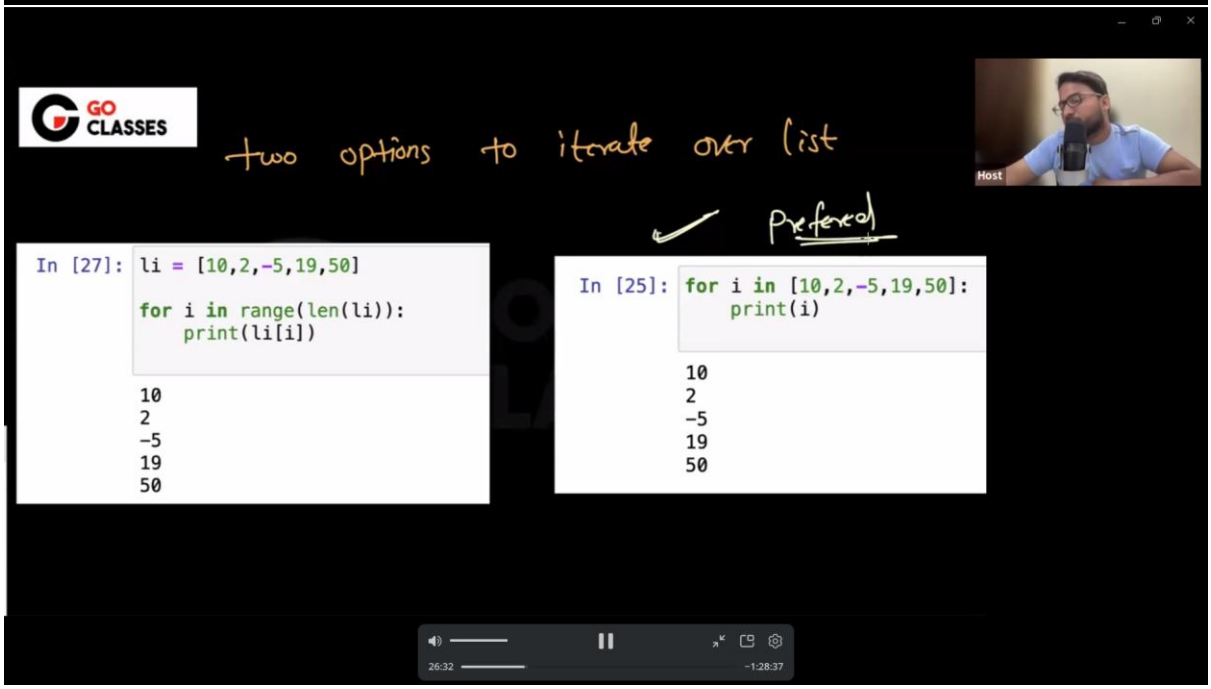

GO CLASSES
range() with One Number
• If range() is given only one number
– It will start counting at 0
– And will count up to (but not including) that number
>>> one = list(range(4))
>>> print(one)
[0, 1, 2, 3]
www.goclasses.in
Host


GO CLASSES
Host
In [14]: list(range(5,10))
Out[14]: [5, 6, 7, 8, 9]
In [15]: list(range(10,5))
Out[15]: []

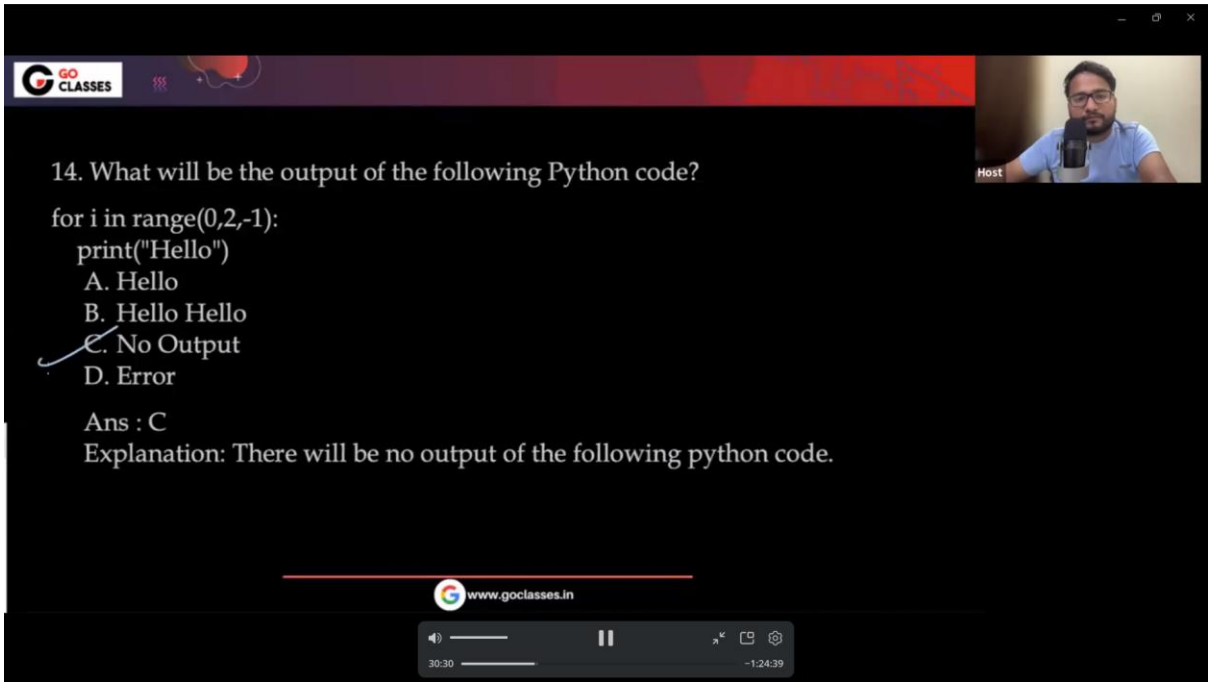

GO CLASSES
Host
In [19]: list(range(5,10,3))
Out[19]: [5, 8]
In [20]: list(range(-10,-5,3))
Out[20]: [-10, -7]
In [21]: list(range(10,5,3))
Out[21]: []
In [22]: list(range(10,5,-3))
Out[22]: [10, 7]
20:56
-1:34:13

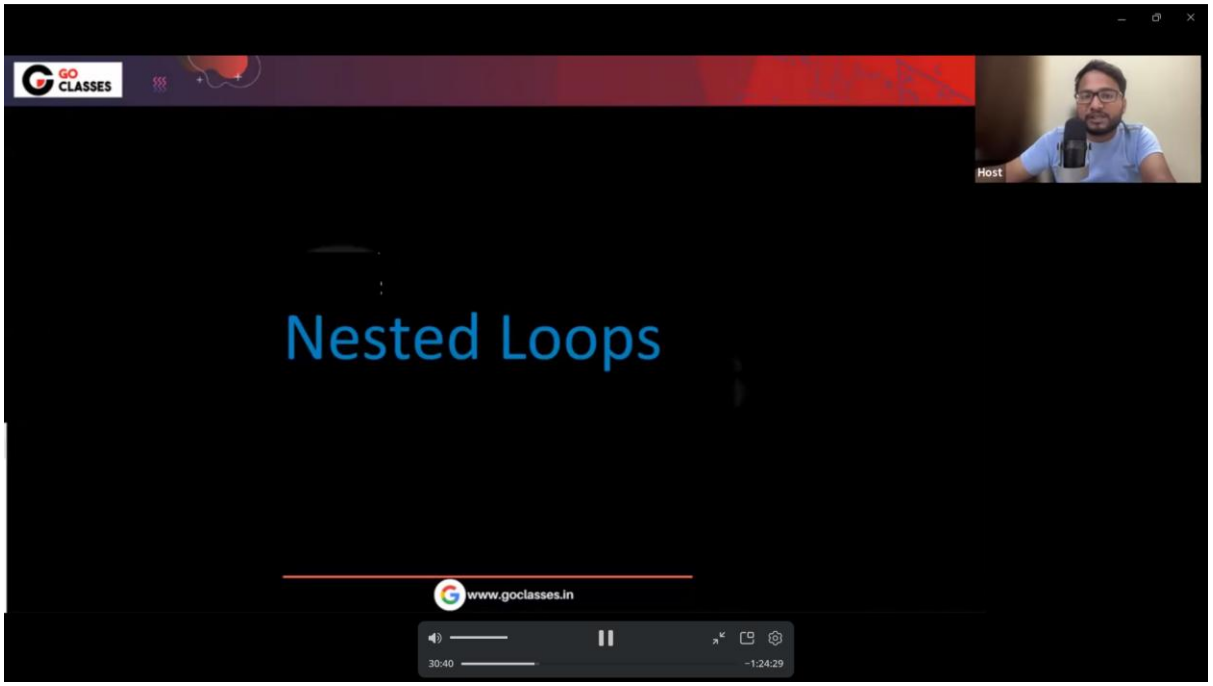

GO CLASSES
Counting Down with range()
Host
• By default, range() counts up
– But we can change this behavior
• If the STEP is set to a negative number, then range() can be used to count down
>>> downA = list(range(10, 0, -1))
>>> print(downA)
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
>>> downB = list(range(18, 5, -2))
>>> print(downB)
[18, 16, 14, 12, 10, 8, 6]
www.goclasses.in

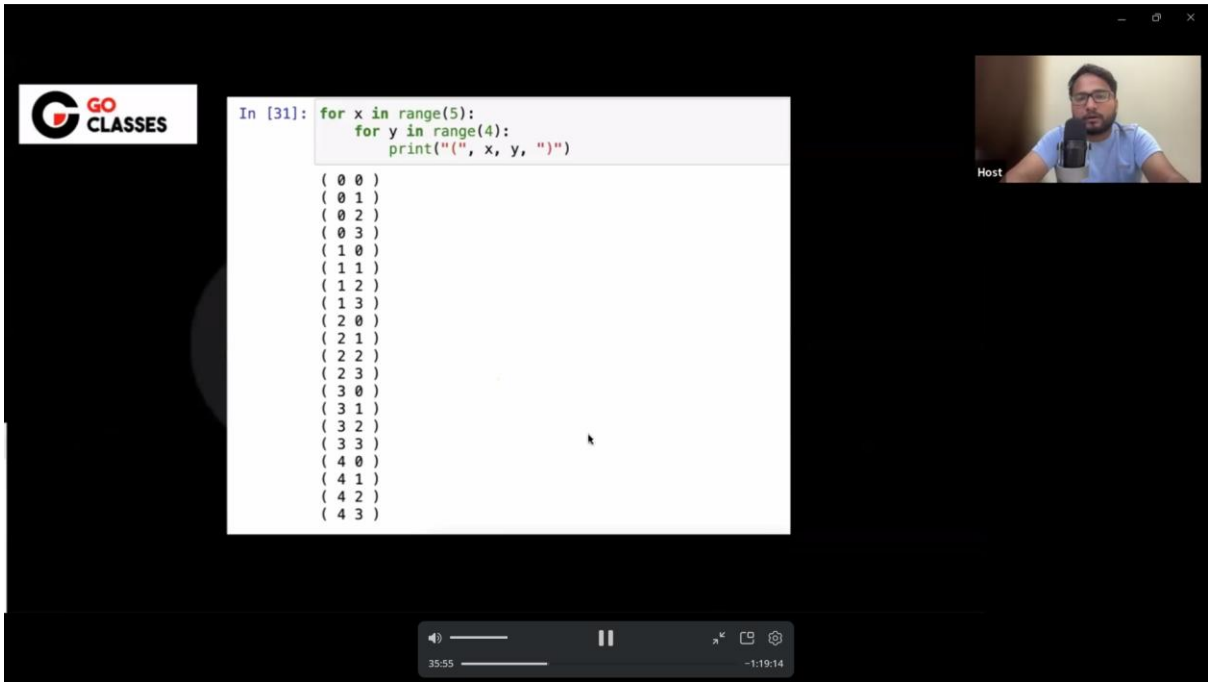

GO CLASSES
Host
• range(a, b) generates a list-like structure containing the numbers from a to b: including a but not including b. For example, range(3, 7) is the sequence 3, 4, 5, 6s. This assumes a is smaller than b; if the reverse is true the list generated is empty.
• range(b) is the same as range(0, b). For example, range(5) is [0, 1, 2, 3, 4].
• range(a, b, c) uses an increment of c to go from one number to the next. So range(2, 12, 3) is [2, 5, 8, 11] and range(5, 0, −1) is [5, 4, 3, 2, 1].
0 is not
www.goclasses.in

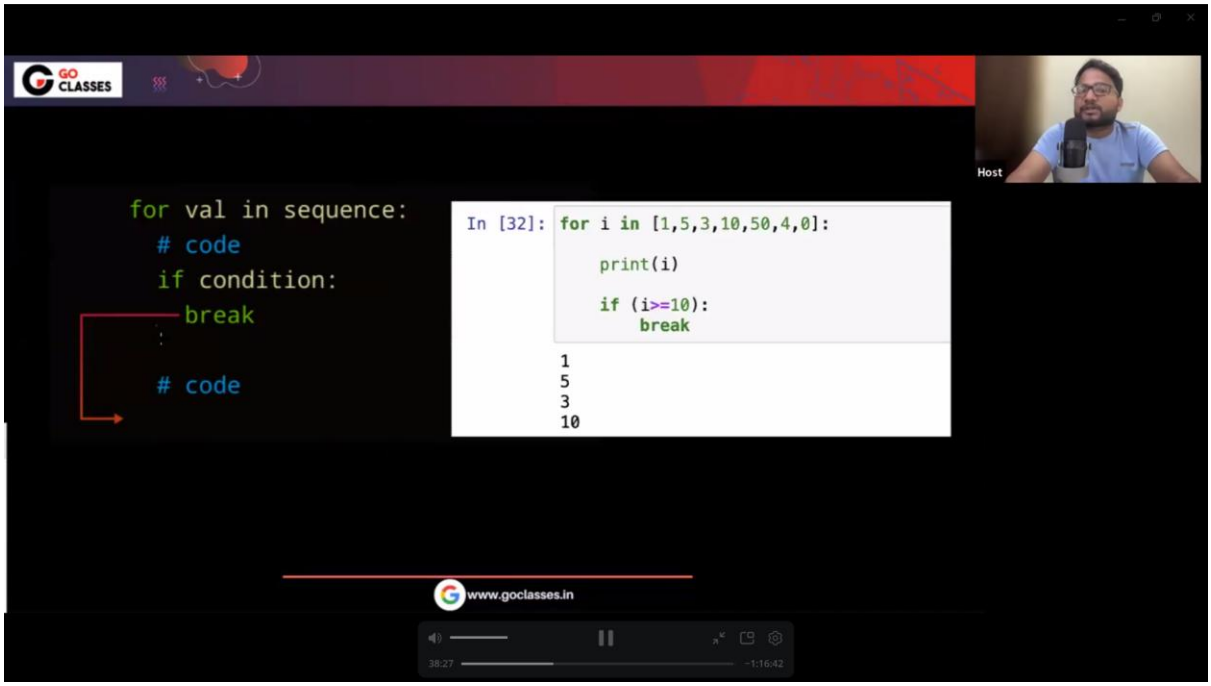

GO CLASSES
Host
two options to iterate over list
Preferred
In [27]:
li = [10,2,-5,19,50]
for i in range(len(li)):
print(li[i])
10
2
-5
19
50
In [25]:
for i in [10,2,-5,19,50]:
print(i)
10
2
-5
19
50
26:32
-1:28:37

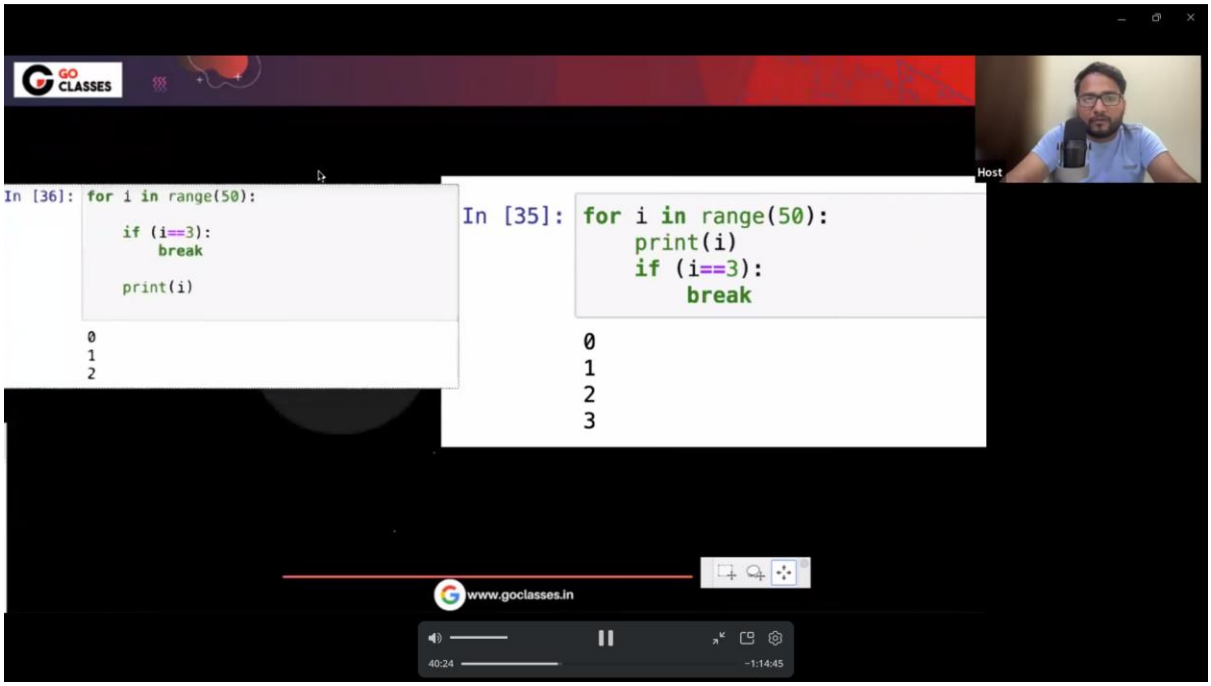

GO CLASSES
Host
In [29]:
for i in ["apple", "bye", "good", 1, [4,6]]:
print(i)
apple
bye
good
1
[4, 6]
29:06
-1:26:03

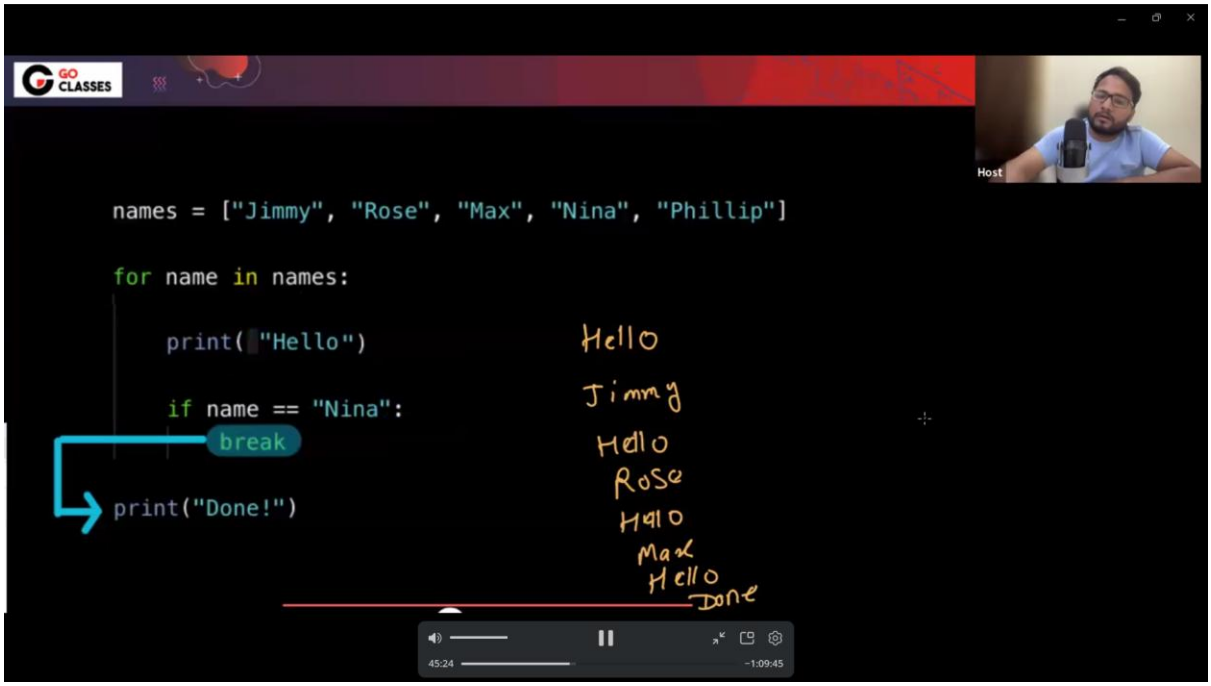

GO CLASSES
Host
14. What will be the output of the following Python code?
for i in range(0,2,-1):
print("Hello")
A. Hello
B. Hello Hello
C. No Output
D. Error
Ans : C
Explanation: There will be no output of the following python code.
www.goclasses.in
30:30
-1:24:39

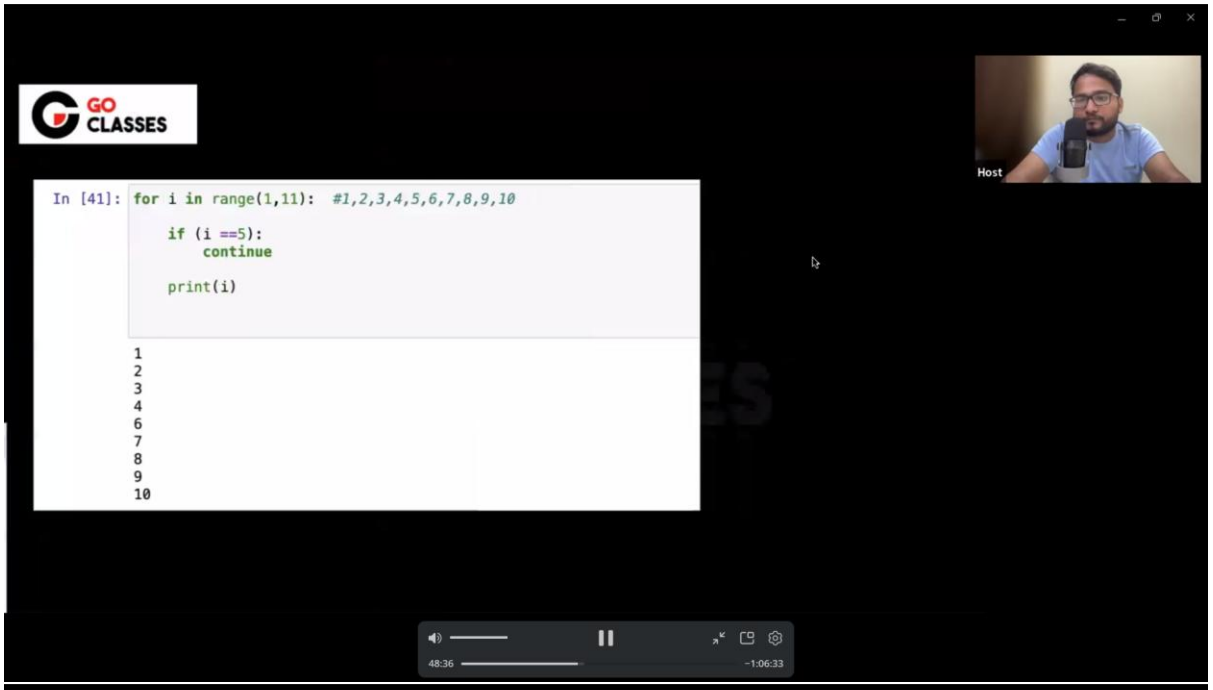

GO CLASSES
Host
Nested Loops
www.goclasses.in
30:40
-1:24:29

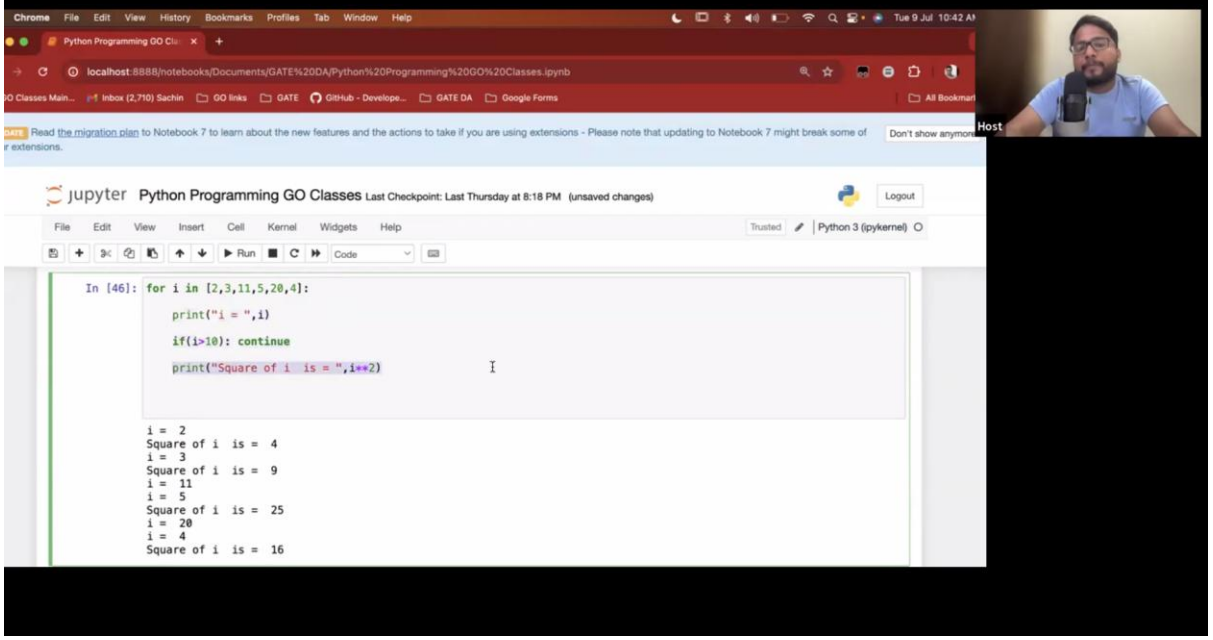

GO CLASSES
Host
In [31]:
for x in range(5):
for y in range(4):
print("(", x, y, ")")
(0 0)
(0 1)
(0 2)
(0 3)
(1 0)
(1 1)
(1 2)
(1 3)
(2 0)
(2 1)
(2 2)
(2 3)
(3 0)
(3 1)
(3 2)
(3 3)
(4 0)
(4 1)
(4 2)
(4 3)
35:55
-1:19:14

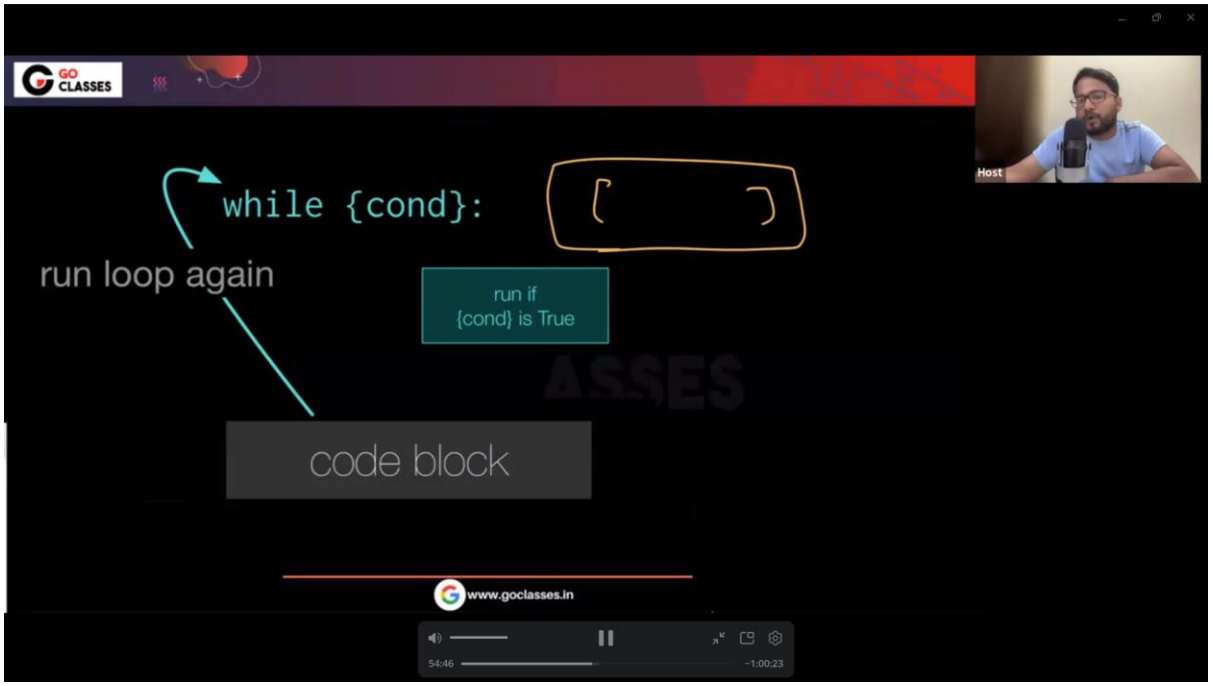

GO CLASSES
Host
for val in sequence:
# code
if condition:
break
code
In [32]:
for i in [1,5,3,10,50,4,0]:
print(i)
if (i>=10):
break
1
5
3
10
www.goclasses.in
38:27
-1:16:42


GO CLASSES
Host
In [36]:
for i in range(50):
if (i==3):
break
print(i)
0
1
2
In [35]:
for i in range(50):
print(i)
if (i==3):
break
0
1
2
3
www.goclasses.in
40:24
-1:14:45

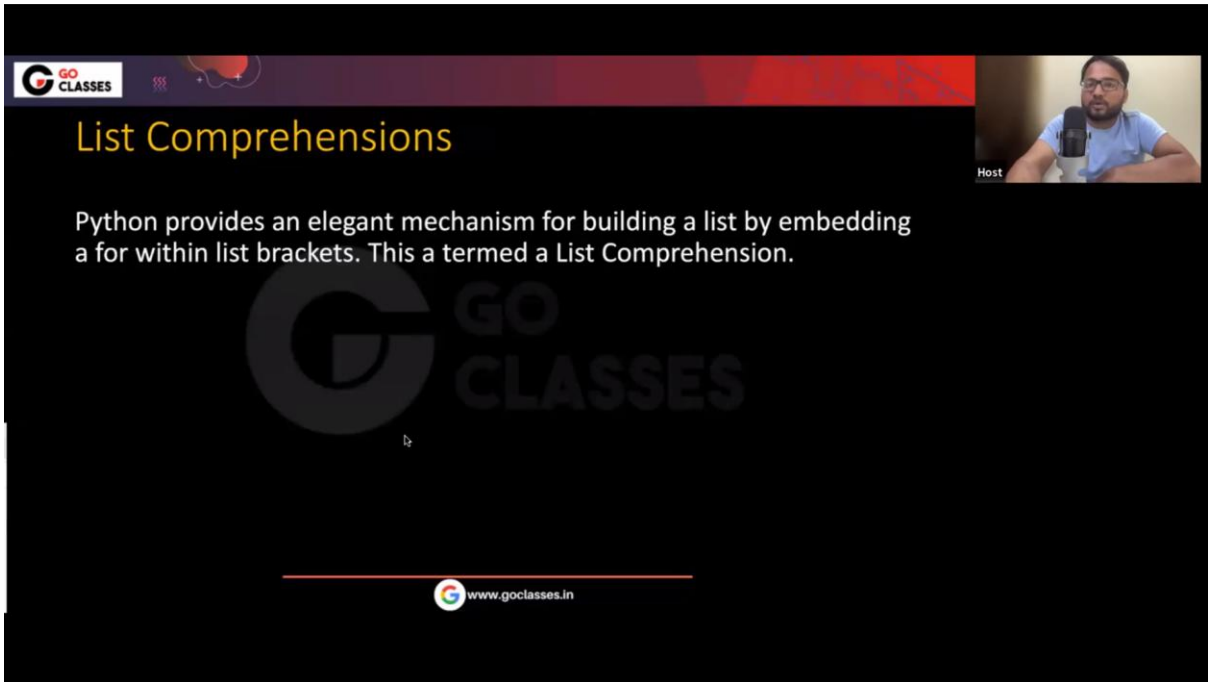

GO CLASSES
Host
names = ["Jimmy", "Rose", "Max", "Nina", "Phillip"]
for name in names:
print("Hello")
if name == "Nina":
break
print("Done!")
Hello
Jimmy
Hello
Rose
Hello
Max
Hello
Done
45:24
-1:09:45

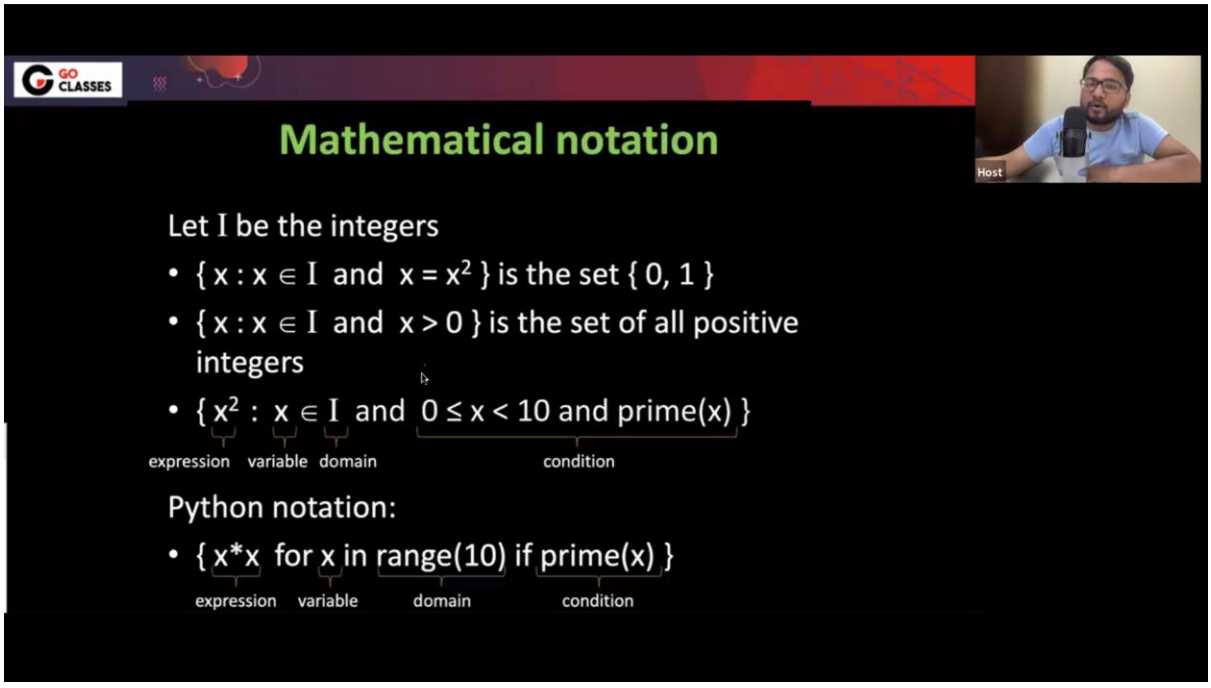

GO CLASSES
Host
In [41]:
for i in range(1,11): #1,2,3,4,5,6,7,8,9,10
if (i ==5):
continue
print(i)
1
2
3
4
6
7
8
9
10
48:36
-1:06:33

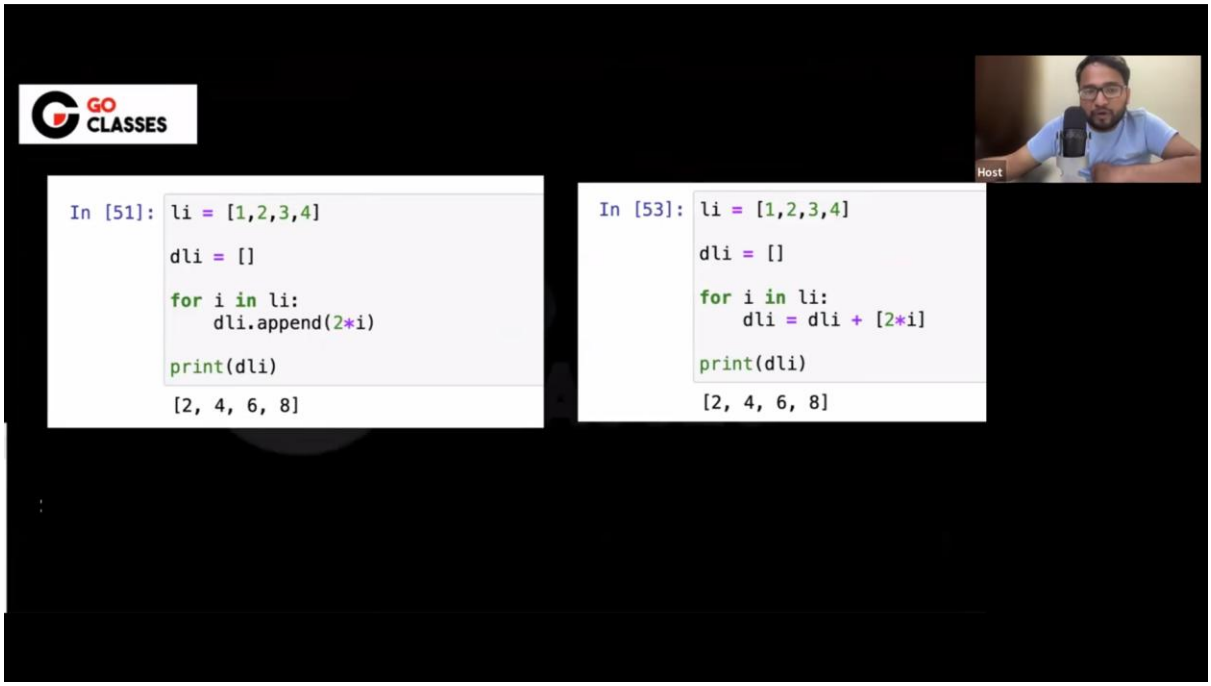

Host
localhost:8888/notebooks/Documents/GATE%20DA/Python%20Programming%20GO%20Classes.ipynb
jupyter Python Programming GO Classes Last Checkpoint: Last Thursday at 8:18 PM (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [46]:
for i in [2,3,11,5,20,4]:
print("i = ",i)
if(i>10): continue
print("Square of i is = ",i**2)
i = 2
Square of i is = 4
i = 3
Square of i is = 9
i = 11
i = 5
Square of i is = 25
i = 20
i = 4
Square of i is = 16


GO CLASSES
while {cond}:
run loop again
run if
{cond} is True
code block
Host
www.goclasses.in
54:46
-1:00:23

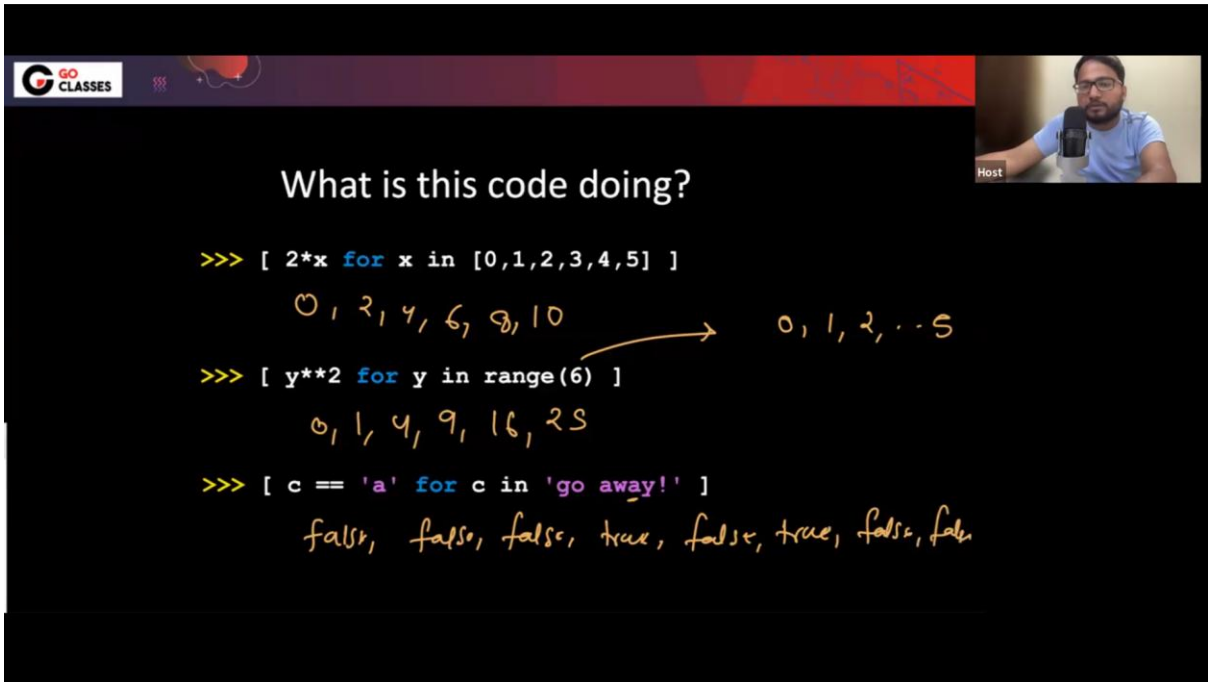

GO CLASSES
Host
i = 0
while i < 10:
print(i)
i = i + 1
print('All done')
www.goclasses.in

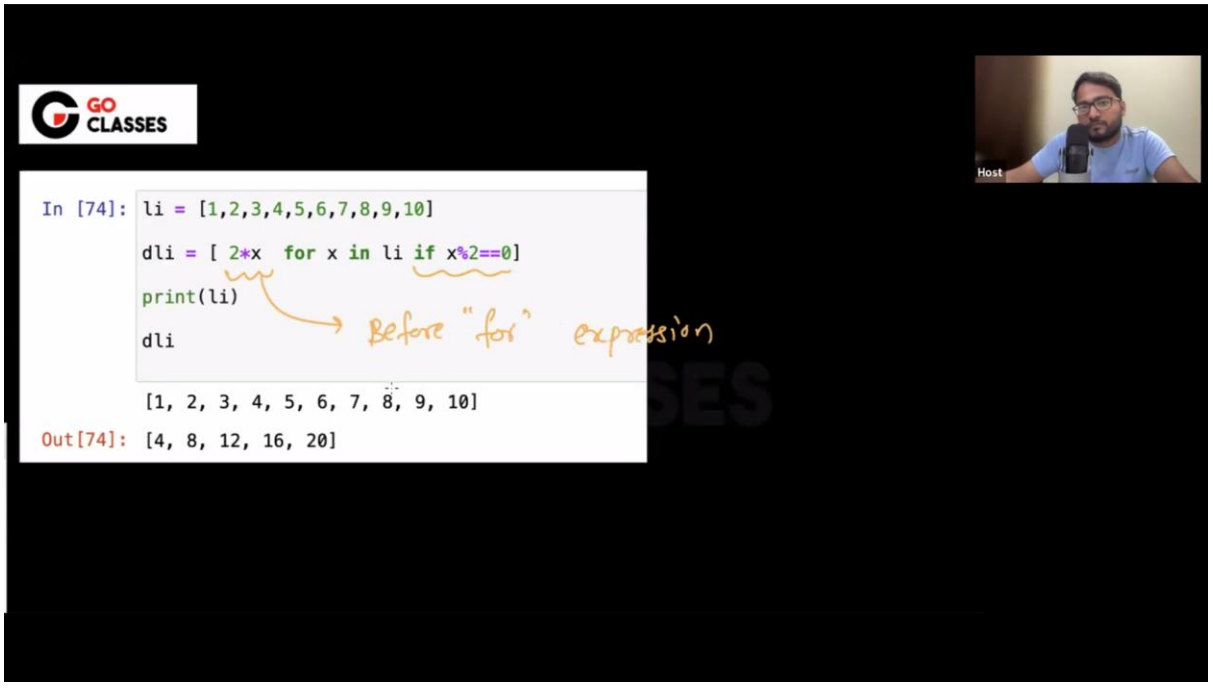

GO CLASSES
List Comprehensions
Python provides an elegant mechanism for building a list by embedding a for within list brackets. This a termed a List Comprehension.
GO CLASSES
www.goclasses.in
Host

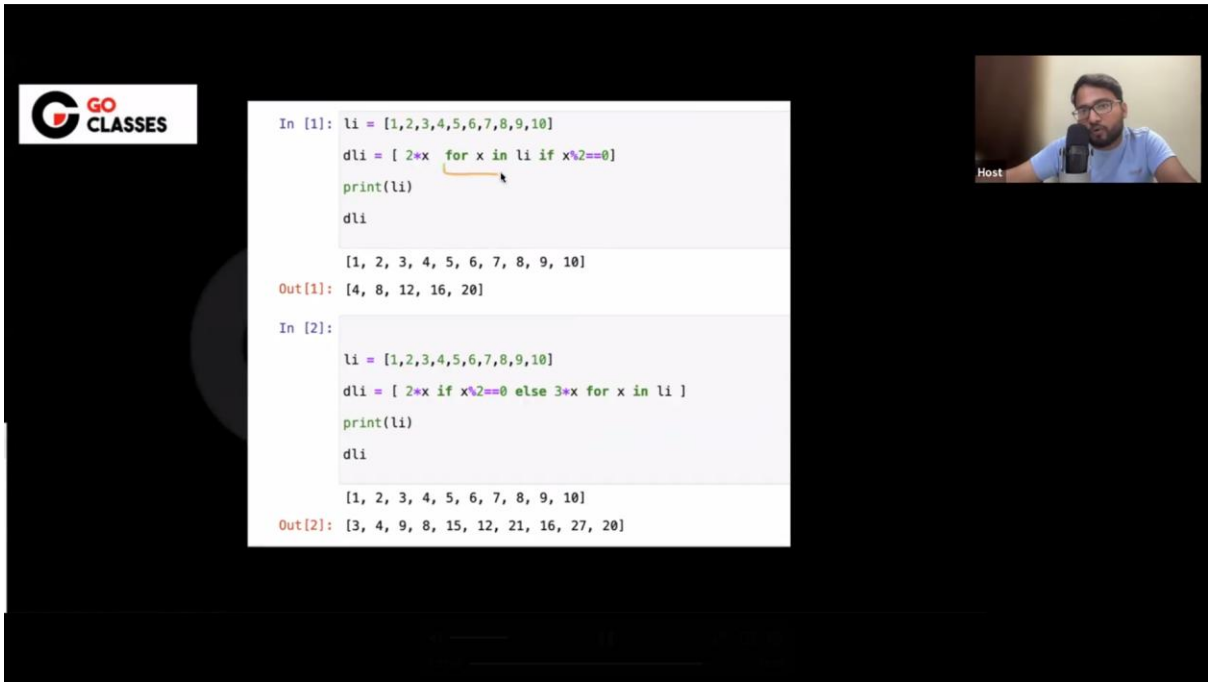

GO CLASSES
Mathematical notation
Let I be the integers
• { x : x ∈ I and x = x² } is the set { 0, 1 }
• { x : x ∈ I and x > 0 } is the set of all positive integers
• { x² : x ∈ I and 0 ≤ x < 10 and prime(x) }
expression variable domain condition
Python notation:
• { x*x for x in range(10) if prime(x) }
expression variable domain condition
Host


GO CLASSES
Host
In [51]:
li = [1,2,3,4]
dli = []
for i in li:
dli.append(2*i)
print(dli)
[2, 4, 6, 8]
In [53]:
li = [1,2,3,4]
dli = []
for i in li:
dli = dli + [2*i]
print(dli)
[2, 4, 6, 8]


GO CLASSES
Host
In [58]:
li = [1,2,3,4]
dli = []
for i in li:
dli.append(2*i)
print(dli)
[2, 4, 6, 8]
In [56]:
li = [1,2,3,4]
dli = [2*x for x in li]
dli
Out[56]: [2, 4, 6, 8]
1:10:10
-44:59


GO CLASSES
Host
What is this code doing?
>>> [2*x for x in [0,1,2,3,4,5]]
0, 2, 4, 6, 8, 10
0, 1, 2, .. 5
>>> [y**2 for y in range(6)]
0, 1, 4, 9, 16, 25
>>> [c == 'a' for c in 'go away!']
false, false, false, true, false, true, false, fals


GO CLASSES
Host
In [74]:
li = [1,2,3,4,5,6,7,8,9,10]
dli = [2*x for x in li if x%2==0]
print(li)
dli
Before "for" expression
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Out[74]: [4, 8, 12, 16, 20]



before loop only if wont work