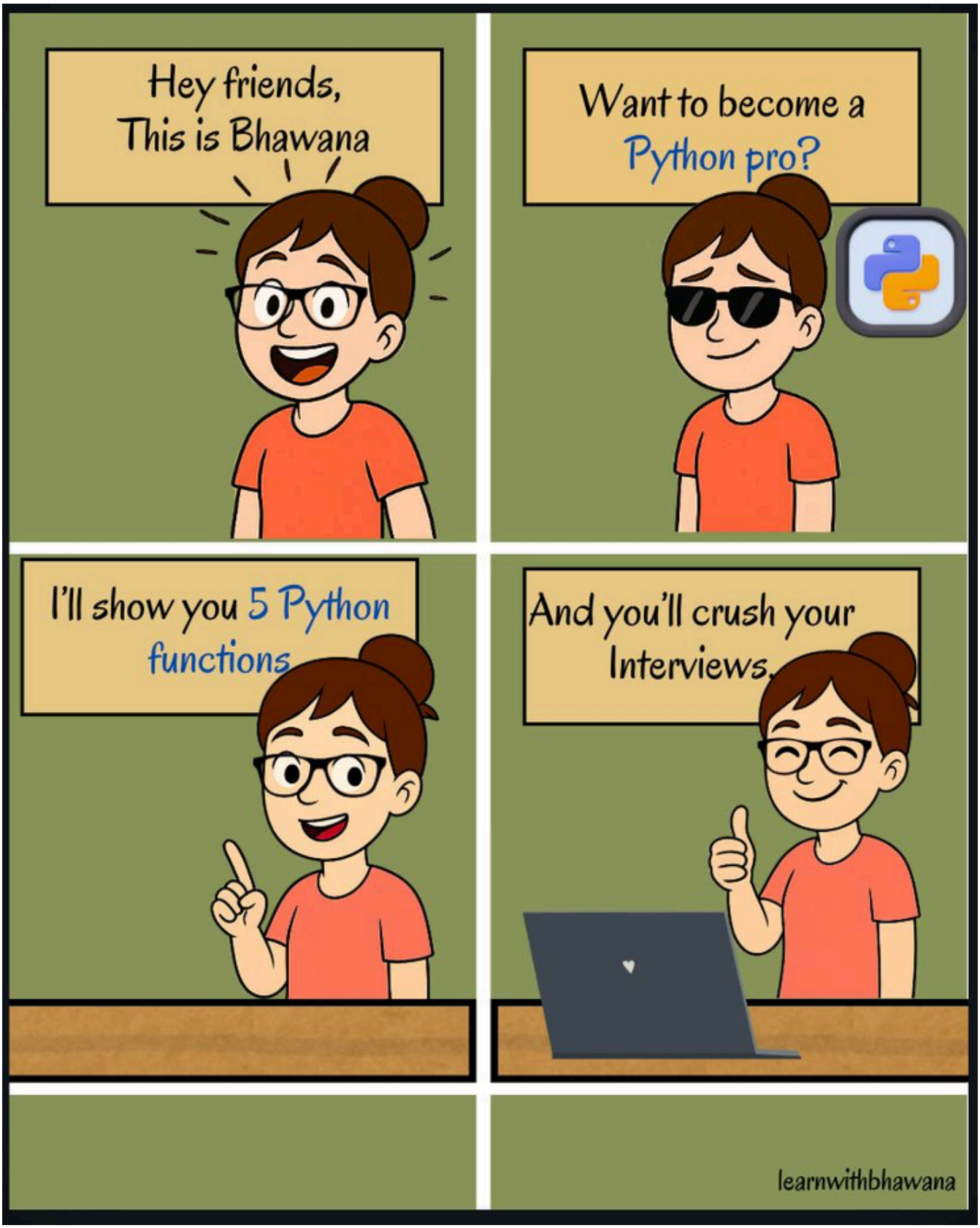
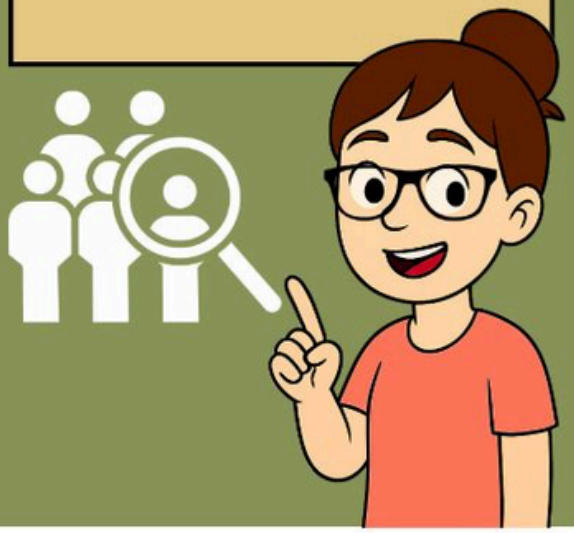
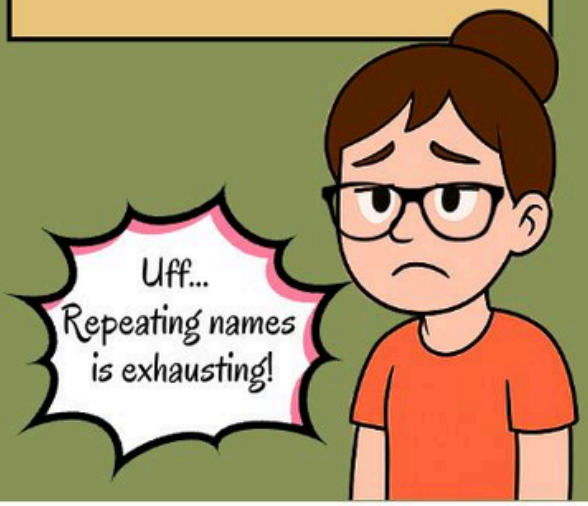

Hey friends,
This is Bhawana
Want to become a
Python pro?
I'll show you 5 Python
functions
And you'll crush your
Interviews.
learnwithbhawana

Imagine I'm scheduling interviews for 10 candidates.



Instead of writing names again and again,

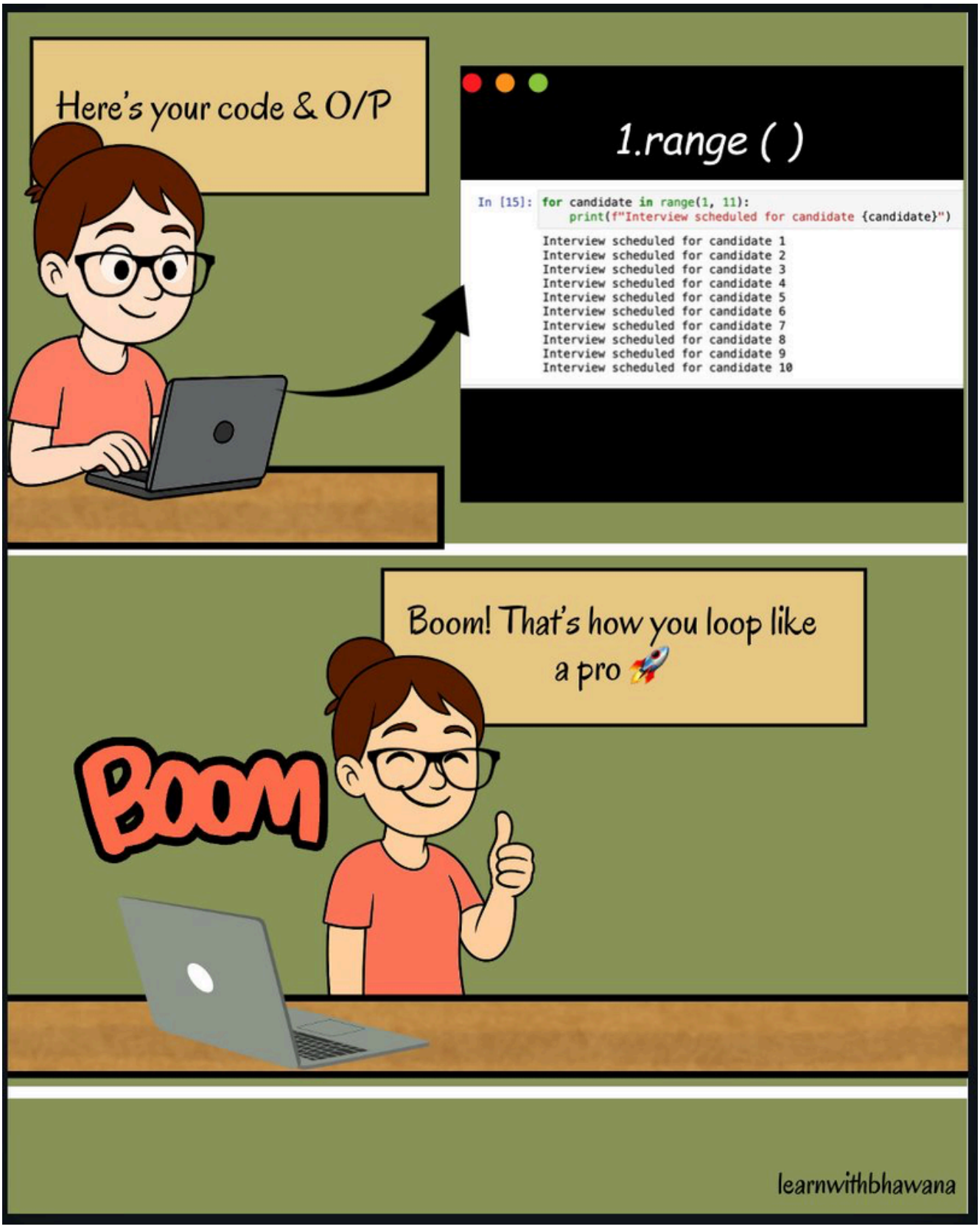


Let's use," range function"

range ( )

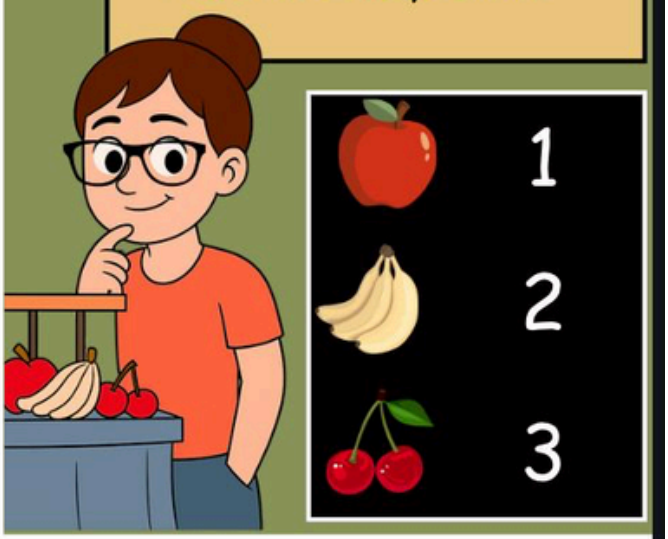
Syntax: range(start, stop, step)

- start → number to begin (default = 0)
- stop → number to end (exclusive)
- step → jump size (default = 1)


Here's your code & O/P
1.range ( )
In [15]:
for candidate in range(1, 11):
print(f"Interview scheduled for candidate {candidate}")
Interview scheduled for candidate 1
Interview scheduled for candidate 2
Interview scheduled for candidate 3
Interview scheduled for candidate 4
Interview scheduled for candidate 5
Interview scheduled for candidate 6
Interview scheduled for candidate 7
Interview scheduled for candidate 8
Interview scheduled for candidate 9
Interview scheduled for candidate 10
Boom! That's how you loop like a pro
BOOM
learnwithbhawana

Ever been at a fruit stall ??

and needed both the fruit AND its position?



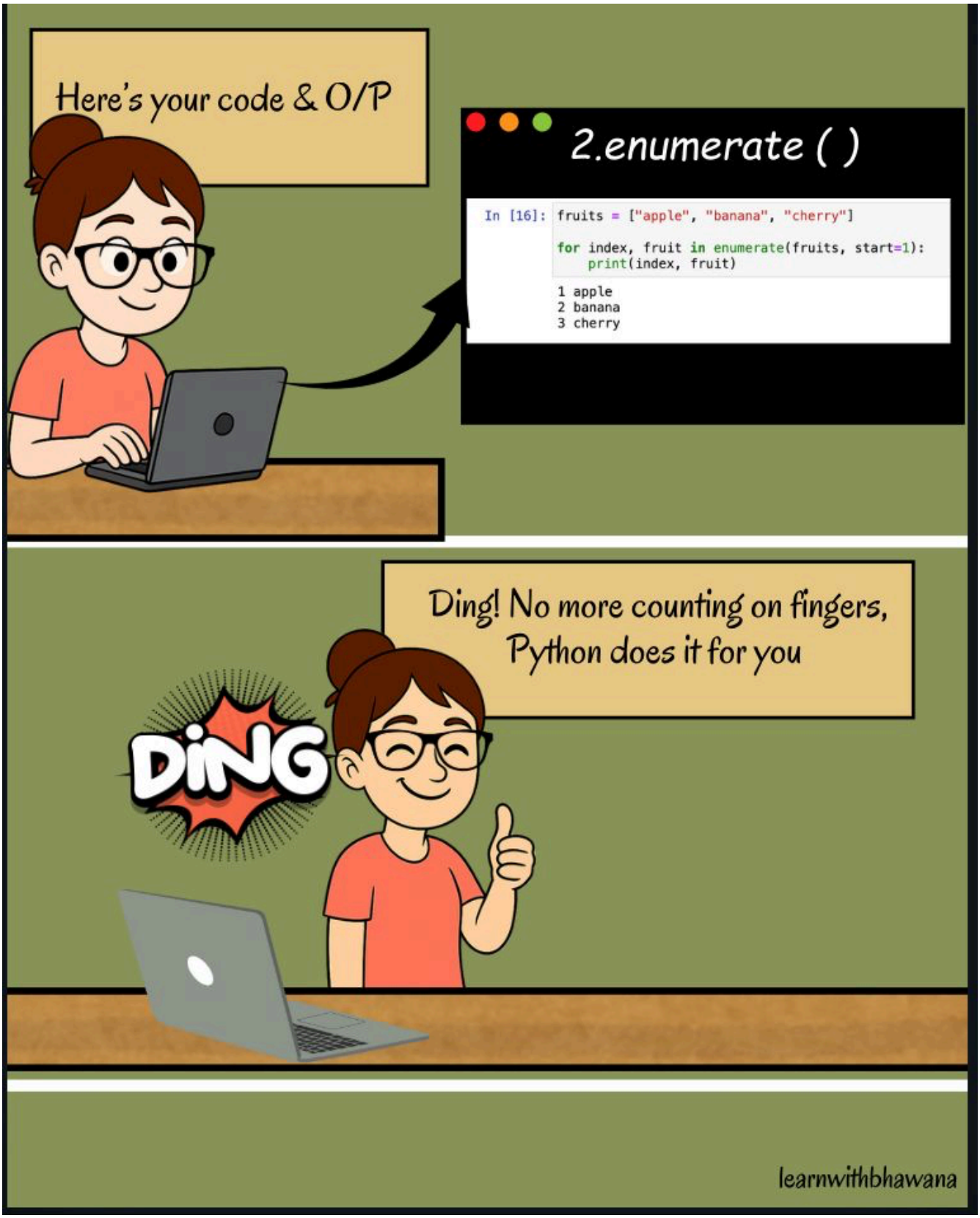
"enumerate function" ,does exactly that.

enumerate( )

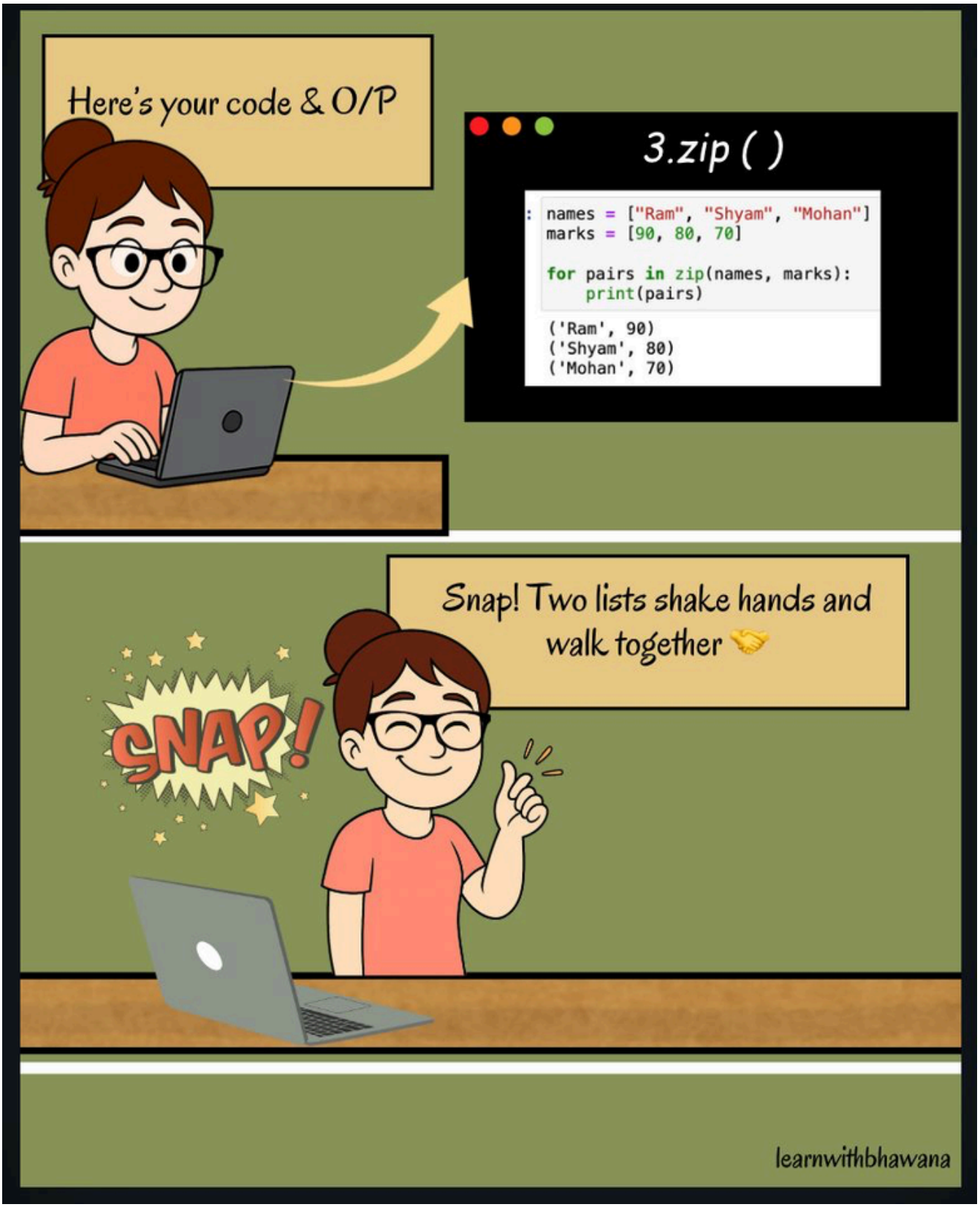
Syntax: enumerate(iterable, start=0)

- iterable → sequence to loop (list, tuple, string)
- start → index to begin count (default = 0)

learnwithbhawana


Here's your code & O/P
2.enumerate ( )
In [16]:
fruits = ["apple", "banana", "cherry"]
for index, fruit in enumerate(fruits, start=1):
print(index, fruit)
1 apple
2 banana
3 cherry
Ding! No more counting on fingers, Python does it for you
DiNG
learnwithbhawana


Looking to pair names with their scores ??
90
70
RAM
SHYAM
90
70
Like ,making perfect pairs from two lists
["Ram","Shyam","Mohan"]
[90,80,70]
"zip function" matches them side by side
zip( )
Syntax: zip(iterable1, iterable2, ...)
• iterable1, iterable2: two lists/tuples (or more)
• zip(): pairs elements index-wise → forms tuples (1st↔1st, 2nd↔2nd...)
learnwithbhawana


Here's your code & O/P
3.zip ( )
: names = ["Ram", "Shyam", "Mohan"]
marks = [90, 80, 70]
for pairs in zip(names, marks):
print(pairs)
('Ram', 90)
('Shyam', 80)
('Mohan', 70)
Snap! Two lists shake hands and walk together 🤝
SNAP!
learnwithbhawana

Think of a kitchen strainer

It lets only pasta through and holds back water.

filter function picks only what meets your condition

filter( )

Syntax: filter(function, iterable)

- function → condition/test (returns True/False)
- iterable → list, tuple, etc.
- Result → only elements passing condition

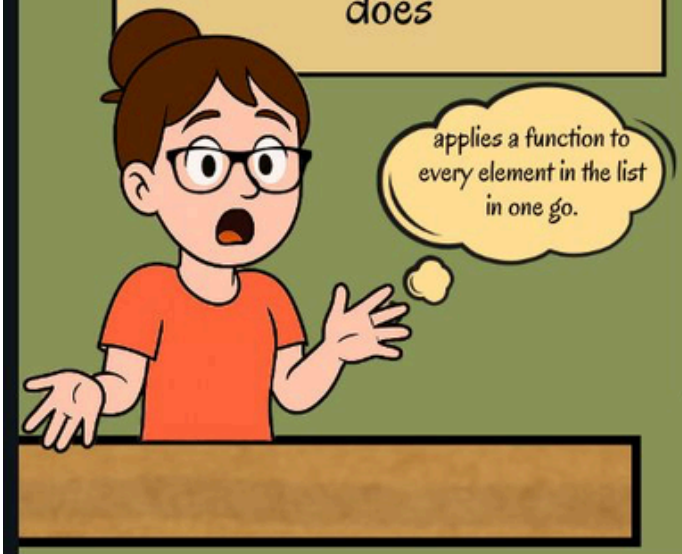
learnwithbhawana


Here's your code & O/P
4.filter ( )
nums = [1, None, 3, 4, None, 6]
valid_nums = filter(lambda x: x is not None, nums)
print(list(valid_nums))
[1, 3, 4, 6]
SWIPE LEFT!
Swipe left! Bad data gone, only the good ones stay ✅
learnwithbhawana

Imagine you're in a juice shop

You give a list of fruits, and the juicer applies the same process (squeeze) to each fruit in one go.

That's what "map function" does

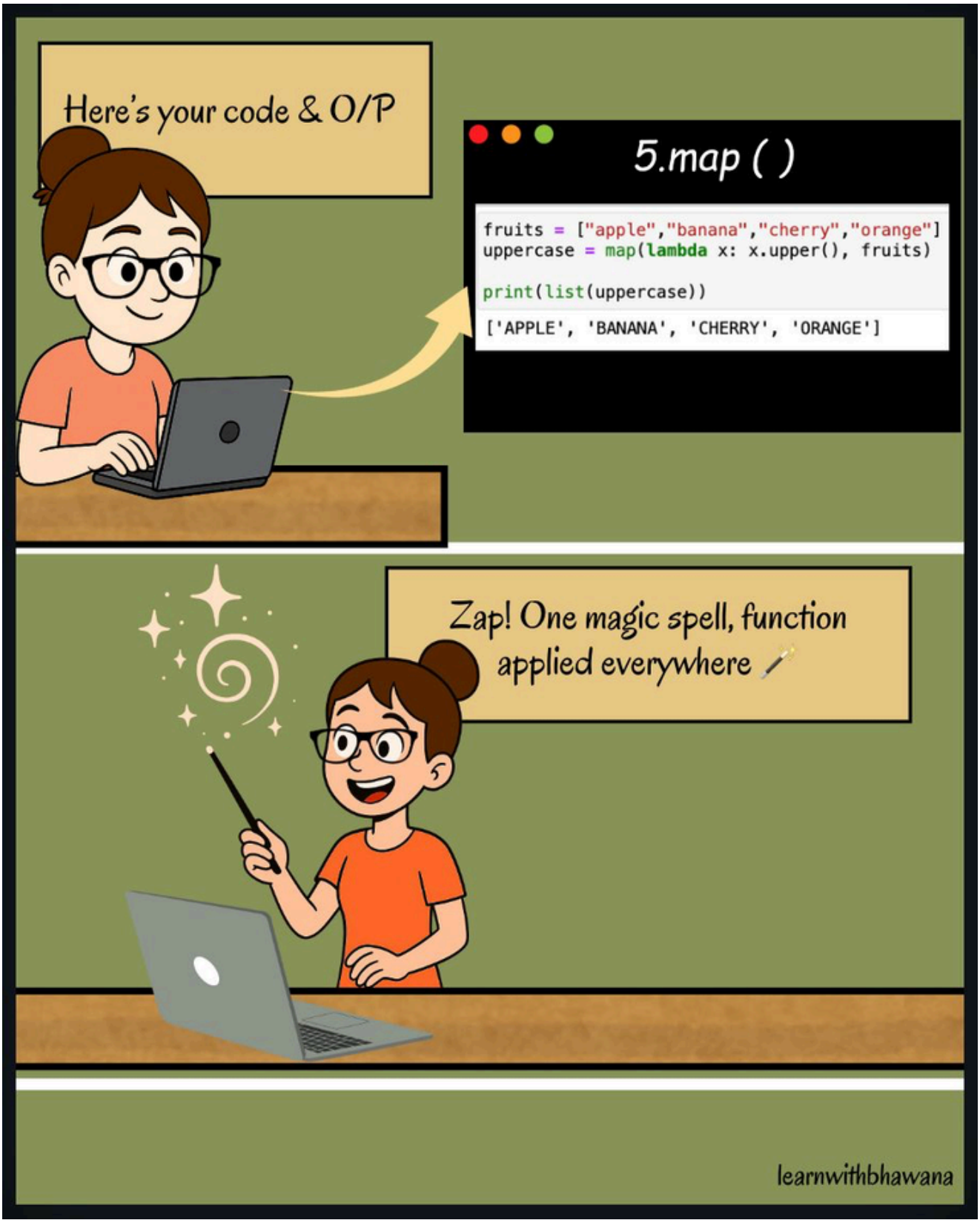


map( )

Syntax: map(function, iterable)

- function → operation to apply (e.g., square, uppercase)
- iterable → list, tuple, etc.
- Result → new iterable with function applied to each element

learnwithbhawana


Here's your code & O/P
5.map ( )
fruits = ["apple","banana","cherry","orange"]
uppercase = map(lambda x: x.upper(), fruits)
print(list(uppercase))
['APPLE', 'BANANA', 'CHERRY', 'ORANGE']
Zap! One magic spell, function applied everywhere
learnwithbhawana


Hope you had fun learning!
More comic-style Python hacks coming your way
PYTHON
HACK
Save, follow, and share with your coder friends!
Save This
SUBSCRIBE NOW
Share
learnwithbhawana