// FOR + FOR — nested loops

```
for i in range(1, 4):
    for j in range(1, 4):
        print(f"{i} x {j} = {i * j}")

# inner completes fully before outer moves
# total iterations = outer x inner
```

Each level adds 4 spaces. Use different variable names.

// FOR + IF — filter while you iterate

```
word = "RedHorn"
vowels = "aeiou"

for letter in word:
    if letter.lower() in vowels:
        print(f"{letter} — vowel")

# if at outer, inner, or both levels
```

The loop visits everything. if decides what to do.

// ACCUMULATOR IN NESTED LOOPS

```
count = 0              # before outer loop

for word in words:
    for letter in word:
        if letter in vowels:
            count += 1 # inside inner loop

print(count)           # after outer loop
```

Define before. Update inside. Use after.

// WHILE + FOR — outer controls, inner processes

```
while True:
    word = input('Enter word or quit: ')
    if word == 'quit':
        break
    count = 0
    for letter in word:
        if letter.lower() in "aeiou":
            count += 1
    print(f"Vowels: {count}")
```

while sets the pace. for does the work.

// BREAK IN NESTED LOOPS

```
# break stops only its own loop
for word in words:
    for letter in word:
        if letter == target:
            break    # inner stops, outer continues

# flag to stop both loops
found = False
for word in words:
    for letter in word:
        if letter == target:
            found = True
            break
    if found:
        break
```

break never bubbles up. Use a flag for multiple levels.

// COMMON MISTAKES

```
! Indentation defines which loop owns the line
! break stops only its own loop
! Define accumulators before the outer loop
! Nested loops multiply — keep sequences small
! Use different variable names at each level
```