

Python collections.Counter — Complete Interview & Practice Guide (1–35)

A comprehensive light-theme PDF covering all Counter interview questions, answers, and examples.

1■■■ What is a Counter and why is it used?

Counter is a subclass of dict used for counting hashable objects. It returns a mapping of

Example:

```
from collections import Counter
c = Counter(['a', 'b', 'a', 'c'])
Output: Counter({'a':2, 'b':1, 'c':1})
```

2■■■ Count elements in a list

```
nums = [1,2,3,2,1,2]
from collections import Counter
print(Counter(nums))
# Output: Counter({2:3, 1:2, 3:1})
```

3■■■ Count characters in a string

```
word = "engineering"
from collections import Counter
print(Counter(word))
# Output: Counter({'e':3, 'n':3, 'g':2, 'i':2, 'r':1})
```

4■■■ Count words in a sentence (case-insensitive)

```
sentence = "Cat dog CAT Dog cat"
from collections import Counter
print(Counter(sentence.lower().split()))
# Output: Counter({'cat':3, 'dog':2})
```

5■■■ Find top-K frequent elements

```
nums = [1,1,2,3,3,3,4,4,4,4]
from collections import Counter
print(Counter(nums).most_common(2))
# Output: [(4,4), (3,3)]
```

6■■■ Reconstruct from Counter

```
c = Counter({'a':2, 'b':2, 'c':1})
print(''.join(c.elements()))
# Output: aabbc
```

7■■■ Perform arithmetic on Counters

```
a = Counter("apple")
b = Counter("pear")
print(a+b) # Counter({'p':3, 'a':2, 'e':2, 'l':1, 'r':1})
```

```
print(a-b) # Counter({'p':1,'l':1})
```

8 Update counts dynamically

```
c = Counter()
c.update("abc")
c.update("ab")
print(c) # Counter({'a':2,'b':2,'c':1})
```

9 Count only specific items (conditional)

```
text = "This is a sample sentence"
vowels = set("aeiou")
from collections import Counter
print(Counter(ch for ch in text.lower() if ch in vowels))
# Output: Counter({'e':4,'a':2,'i':2})
```

Intersection and union

```
a = Counter("banana")
b = Counter("bandana")
print(a & b) # Counter({'a':3,'n':2,'b':1})
print(a | b) # Counter({'a':3,'n':2,'b':1,'d':1})
```

11 Clean up zero/negative counts

```
c = Counter(a=3,b=-1,c=0)
c += Counter()
print(c) # Counter({'a':3})
```

12 Check anagrams

```
def is_anagram(a,b):
    from collections import Counter
    return Counter(a)==Counter(b)
print(is_anagram("listen","silent")) # True
print(is_anagram("apple","pale")) # False
```

13 Detect duplicates

```
nums = [1,2,2,3,3,3,4]
from collections import Counter
print([n for n,c in Counter(nums).items() if c>1])
# Output: [2,3]
```

14 Group by frequency

```
nums = [1,1,2,2,2,3]
from collections import Counter
freq = Counter(nums)
grouped = {}
for k,v in freq.items():
    grouped.setdefault(v,[]).append(k)
print(grouped)
# Output: {2:[1],3:[2],1:[3]}
```

15 Find element with odd frequency

```
nums = [1,2,3,2,3,1,3]
from collections import Counter
c = Counter(nums)
print([k for k,v in c.items() if v%2==1]) # [3]
```

16 Count digits in a string

```
from collections import Counter
text = "Account123Balance456"
print(Counter(ch.isdigit() for ch in text))
# Output: Counter({False: 15, True: 6})
```

17 Count uppercase and lowercase letters

```
text = "PythonIsGreat"
print(Counter("Upper" if c.isupper() else "Lower" for c in text))
# Output: Counter({'Lower': 8, 'Upper': 3})
```

18 Count first letters of words

```
sentence = "Data Driven Development Design Decisions"
print(Counter(word[0] for word in sentence.split()))
# Output: Counter({'D': 5})
```

19 Frequency distribution of list elements

```
nums = [1,2,3,4,5,5,4,3,2,1,1]
print(sorted(Counter(nums).items()))
# Output: [(1, 3), (2, 2), (3, 2), (4, 2), (5, 2)]
```

20 Find most frequent number

```
nums = [1,2,3,4,4,2,2,3,4]
c = Counter(nums)
print(max(c, key=c.get)) # Output: 2
```

21 Convert Counter to dictionary

```
nums = [1,1,2,3,3,3]
print(dict(Counter(nums)))
# Output: {1: 2, 2: 1, 3: 3}
```

22 Merge multiple Counters

```
a = Counter("apple")
b = Counter("pear")
c = Counter("peach")
print(a + b + c)
# Output: Counter({'p': 4, 'a': 3, 'e': 3, 'l': 1, 'r': 1, 'c': 1, 'h': 1})
```

23 Subtract frequencies

```
a = Counter("mississippi")
```

```
b = Counter("miss")
a.subtract(b)
print(a)
# Output: Counter({'i': 4, 'p': 2})
```

24■■■ Using += and -= operators

```
c = Counter("abc")
c += Counter("bcd")
print(c) # Counter({'b': 2, 'c': 2, 'a': 1, 'd': 1})
c -= Counter("cd")
print(c) # Counter({'b': 2, 'a': 1})
```

25■■■ Count items by category

```
data = [('A', 10), ('B', 20), ('A', 30), ('B', 10), ('C', 5)]
totals = Counter()
for k, v in data:
    totals[k] += v
print(totals)
# Output: Counter({'A': 40, 'B': 30, 'C': 5})
```

26■■■ Group by word length

```
words = ["hi", "hello", "hey", "good", "morning"]
print(Counter(len(w) for w in words))
# Output: Counter({5: 2, 2: 1, 3: 1, 4: 1})
```

27■■■ Find most common words

```
text = "to be or not to be that is the question"
print(Counter(text.split()).most_common(3))
# Output: [('to', 2), ('be', 2), ('or', 1)]
```

28■■■ Count character bigrams

```
text = "banana"
bigrams = [text[i:i+2] for i in range(len(text)-1)]
print(Counter(bigrams))
# Output: Counter({'an': 2, 'na': 2, 'ba': 1})
```

29■■■ Count overlapping substrings

```
text = "aaaa"
pattern = "aa"
count = sum(1 for i in range(len(text)) if text.startswith(pattern, i))
print(count) # Output: 3
```

30■■■ Check permutation using Counter

```
def is_permutation(a, b):
    return Counter(a) == Counter(b)
print(is_permutation("abcd", "dcba")) # True
```

31■■■ Find missing elements between lists

```
a = [1,2,3,4]
b = [1,2,4,4,5]
missing = Counter(b) - Counter(a)
print(list(missing.elements())) # [4, 5]
```

32■■■ Intersection of two lists

```
list1 = [1,2,2,3]
list2 = [2,2,4]
print(list((Counter(list1) & Counter(list2)).elements()))
# Output: [2, 2]
```

33■■■ Voting system — find winner

```
votes = ["Alice", "Bob", "Alice", "Alice", "Bob"]
winner = Counter(votes).most_common(1)[0]
print("Winner:", winner[0]) # Winner: Alice
```

34■■■ Count HTTP status codes

```
logs = [200, 200, 404, 500, 200, 404]
print(Counter(logs))
# Output: Counter({200: 3, 404: 2, 500: 1})
```

35■■■ Count words starting with prefix

```
words = ["apple", "app", "application", "banana", "bat"]
prefix = "app"
count = sum(1 for w in words if w.startswith(prefix))
print(count) # Output: 3
```

■ Summary

This document includes 35 curated Python Counter questions with detailed answers, code examples, and use cases — from fundamentals to advanced analytics patterns.