Solution 28: Grouping with Dict of Lists

1) What does it print?

```
words = ["cat", "car", "dog", "door"]

for w in words:
    print(w[0])
```

Output:

```
c
c
d
d
```

2) Fill in the blanks (first + last letter)

```
w = "apple"

first = w[0]
last  = w[-1]

print(first)
print(last)
```

Output:

```
a
e
```

3) What does it print? (last letters)

```
words = ["cat", "dog", "car"]

for w in words:
    print(w[-1])
```

Output:

```
t
g
r
```

4) Fill in the blanks (basic grouping)

```
words = ["cat", "car", "dog"]

groups = {}

for w in words:
    key = w[0]
    if key not in groups:
        groups[key] = []
    groups[key].append(w)

print(groups)
```

Expected output:

```
{'c': ['cat', 'car'], 'd': ['dog']}
```

5) What does it print? (trace the dictionary)

```
words = ["ant", "apple", "bat", "ball"]

groups = {}

for w in words:
    key = w[0]
    if key not in groups:
        groups[key] = []
    groups[key].append(w)

print(groups)
```

Output:

```
{'a': ['ant', 'apple'], 'b': ['bat', 'ball']}
```

6) Fix the bug (missing bucket)

One correct fix:

```
words = ["cat", "car", "dog"]
groups = {}

for w in words:
    key = w[0]
    if key not in groups:
        groups[key] = []
    groups[key].append(w)

print(groups)
```

7) Fill in the blanks (group by last letter)

```
words = ["cat", "mat", "car", "bar"]

groups = {}

for w in words:
    key = w[-1]
    if key not in groups:
        groups[key] = []
    groups[key].append(w)

print(groups)
```

Expected output:

```
{'t': ['cat', 'mat'], 'r': ['car', 'bar']}
```

8) Write code (group by word length)

One possible answer:

```
words = ["hi", "cat", "door", "a", "sun", "tree"]

groups = {}

for w in words:
    key = len(w)
    if key not in groups:
        groups[key] = []
    groups[key].append(w)

print(groups)
```

Expected output:

```
{2: ['hi'], 3: ['cat', 'sun'], 4: ['door', 'tree'], 1: ['a']}
```

9) Write code (group numbers: even vs odd)

One possible answer:

```
nums = [5, 2, 7, 4, 1, 6]

groups = {"odd": [], "even": []}

for x in nums:
    if x % 2 == 0:
        groups["even"].append(x)
    else:
        groups["odd"].append(x)

print(groups)
```

Expected output:

```
{'odd': [5, 7, 1], 'even': [2, 4, 6]}
```

10) What does it print? (look inside one bucket)

```
words = ["cat", "car", "dog", "door", "apple"]

groups = {}

for w in words:
    key = w[0]
    if key not in groups:
        groups[key] = []
    groups[key].append(w)

print(groups["d"])
```

Output:

```
['dog', 'door']
```

11) Fill in the blanks (print bucket sizes)

```
words = ["cat", "car", "dog", "door", "apple"]

groups = {}

for w in words:
    key = w[0]
    if key not in groups:
        groups[key] = []
    groups[key].append(w)

for key, bucket in groups.items():
    print(key, "has", len(bucket), "words")
```

Expected output:

```
c has 2 words
d has 2 words
a has 1 words
```

12) Sort each bucket

```
words = ["car", "cat", "door", "dog"]

groups = {}

for w in words:
    key = w[0]
    if key not in groups:
        groups[key] = []
    groups[key].append(w)

for key in groups:
    groups[key].sort()

print(groups)
```

Output:

```
{'c': ['car', 'cat'], 'd': ['dog', 'door']}
```

13) Fill in the blanks (turn groups into counts)

```
groups = {'c': ['cat', 'car'], 'd': ['dog']}

counts = {}

for key, bucket in groups.items():
    counts[key] = len(bucket)

print(counts)
```

Output:

```
{'c': 2, 'd': 1}
```

14) Case-insensitive grouping (ignore upper/lowercase)

One possible answer:

```
names = ["Ava", "adam", "Ben", "bobby", "ALICE"]

groups = {}

for name in names:
    key = name[0].lower()
    if key not in groups:
        groups[key] = []
    groups[key].append(name)

print(groups)
```

Expected output:

```
{'a': ['Ava', 'adam', 'ALICE'], 'b': ['Ben', 'bobby']}
```