Worksheet 26: Python Word Frequency

Name: _____ Date: _____

Instructions

- Answer in the blanks.
- For “write code” questions, write **valid Python code** (**no functions needed**).
- For “what does it print” questions, write the **exact** output (line by line).
- If a question says “order may vary”, any correct order is acceptable.
- You may assume all inputs are valid (unless the question says otherwise).

Part A — Review: dictionary traversal + membership

1) Traverse keys

What is the output? (Order may vary.)

```
freq = {"cat": 3, "dog": 1}

for word in freq:
    print(word)
```

Output:

```

```

2) Traverse key-value pairs

What is the output? (Order may vary.)

```
freq = {"cat": 3, "dog": 1}

for word, count in freq.items():
    print(word, count)
```

Output:

```

```

3) `in` checks keys only

What is the output?

```
freq = {"cat": 3, "dog": 1}

print("cat" in freq)
print(3 in freq)
print("bird" in freq)
```

Output:

```

```

4) Build word frequency

Fill in the blanks so it prints the correct dictionary.

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

freq = {}

for w in words:
    if w in freq:
        freq[w] = _____________________________
    else:
        freq[w] = _____________________________

print(freq)
```

Expected output:

```
{'cat': 3, 'dog': 1}
```

5) Build word frequency

What is the output?

```
words = ["A", "B", "A", "C", "B", "B"]

freq = {}

for w in words:
    if w in freq:
        freq[w] = freq[w] + 1
    else:
        freq[w] = 1

print(freq)
```

Output:

```

```

Part B — Counting patterns (write code + fix bugs)

6) Write code: build `freq`

Write Python code to build a word frequency dictionary and print it.

```
words = ["red", "blue", "red", "green", "blue", "red"]

freq = _________________

for w in words:
    # your code here







print(freq)
```

Expected output:

```
{'red': 3, 'blue': 2, 'green': 1}
```

7) Fill in the missing line (increase frequency)

Fill in the blank so it works.

```
words = ["hi", "hi", "bye", "hi"]
freq = {}

for w in words:
    if w in freq:
        ______________________________
    else:
        freq[w] = 1

print(freq)
```

Expected output:

```
{'hi': 3, 'bye': 1}
```

8) Fix the bug (starting count)

The output should be `{'A': 2, 'B': 1}`.

Find and fix the bug.

```
words = ["A", "A", "B"]
freq = {}

for w in words:
    if w in freq:
        freq[w] = freq[w] + 1
    else:
        freq[w] = 0   # bug!

print(freq)
```

Write the corrected line of code:

```

```

9) Total number of words (write code)

Given a frequency dictionary, write code to compute the **sum of word count**.

```
freq = {"to": 2, "be": 2, "or": 1, "not": 1}

total = 0

# complete the code:
```

Expected output:

```
6
```

10) Safe lookup (write code)

Write code to print the frequency of `target`.

- If `target` is not in the dictionary, print `0`.

```
freq = {"cat": 3, "dog": 1}
target = "bird"

# your code here
```

Expected output:

```
0
```

Part C — From text to words (`split`) + counting

11) `split()`

What is the output?

```
text = "to be or not to be"
words = text.split()
print(words)
```

Output:

```

```

12) Split + count

What is the output?

```
text = "to be or not to be"
words = text.split()

freq = {}

for w in words:
    if w in freq:
        freq[w] = freq[w] + 1
    else:
        freq[w] = 1

print(freq)
```

Output:

```

```

13) Case-insensitive counting (fill in the blanks)

We want `"To"`, `"to"`, and `"TO"` to count as the **same** word.

Hint:

- If `s` is a string, then `s.lower()` returns the lowercase.
- For example, if `s = "AbcD"`, then `l = s.lower()` is `"abcd"`.

Fill in the blanks.

```
words = ["To", "to", "TO", "be"]

freq = {}

for w in words:
    w = _____________________________   # make it lowercase
    if w in freq:
        freq[w] = freq[w] + 1
    else:
        freq[w] = 1

print(freq)
```

Expected output:

```
{'to': 3, 'be': 1}
```

14) Remove punctuation (fill in the blanks)

We want to count words, but ignore commas `,` and periods `.`.

Hint:

- If `s` is a string, then `text.replace(c1, c2)` replaces all the characters of `c1` by `c2`.
- Examples:
  - If `s = "ab cd"`, then `text.replace(" ", "")` replaces space by empty. It returns `"abcd"` without space.
  - If `s = "ab ac"`, then `text.replace("a", "x")` replaces `"a"` by `"x"`. It returns `"xb xc"`.

Fill in the blanks.

```
text = "hi, hi. bye, hi."
text = text.replace(_____________________, _____________________)  # replace commas
text = text.replace(_____________________, _____________________)  # replace period

words = text.split()
print(words)
```

Expected output:

```
['hi', 'hi', 'bye', 'hi']
```

Part D — Most frequent word

15) Find the most frequent word (fill in the blanks)

Fill in the blanks so it prints one most frequent word and its count.

```
freq = {"to": 2, "be": 2, "or": 1, "not": 1}

best_word = ""
best_count = 0

for word, count in freq.items():
    if count > best_count:
        best_word = _____________________________
        best_count = _____________________________

print(best_word, best_count)
```

Expected output (one possible answer):

```
to 2
```

16) Write code: most frequent word from a list

Write code to:

1) build a frequency dictionary
2) print the most frequent word and its count

If there is a tie, print any one.

```
words = ["sun", "moon", "sun", "star", "moon", "sun"]

# count word frequency
freq = ____________________  # create an empty dictionary
for w in words:
    if ____________________:
        freq[w] = freq[w] + 1
    else:
        freq[w] = ____________

# find the most frequent word and its count
best_word = ""
best_count = 0

for word, count in freq.items():
    if ____________________________:
        best_word = word
        best_count = count

print(best_word, best_count)
```

Example output:

```
sun 3
```