P5. Dict is natural for counting. Write a function named `word_freq` to take a string as its argument, count occurrences of each word, and return the count as a dict. Discard any word of length 1 or less, e.g., *"Our class is a programming class."* should be counted to: `{'Our': 1, 'class': 2, 'is': 1, 'programming': 1}`. There will be no count for `'a'` (too short).

Use the P5 template. (`WordCount_Template.py`; note: template is only to encourage intended learning skills and allow smooth auto-grading.)

Example

When invoked by

```
txt = "Evil is done by oneself; " + \
        "by oneself is one defiled. \n " + \
        "Evil is left undone by oneself; " + \
        "by oneself is one cleansed. \n " +  \
        "Purity and impurity are one's own doing. \n" + \
        "No one purifies another. \n" + \
        "No other purifies one."
        # excerpt from Attavagga: Self, www.accesstoinsight.org
print(txt)
wf = word_freq(txt)
print('\nCount:')
print(wf)
```

it results

```
======================================================================
```

```
Evil is done by oneself; by oneself is one  defiled.

 Evil is left undone by oneself; by oneself is one cleansed.

 Purity and impurity are one's own doing.

No one purifies another.

No other purifies one.


Count:

{'doing': 1, 'another': 1, 'purifies': 2, 'own': 1, 'undone': 1, 'is':
4, 'other': 1, 'impurity': 1, 'left': 1, 'by': 4, 'oneself': 4, 'No':
2, 'Purity': 1, 'one': 4, 'cleansed': 1, 'and': 1, 'are': 1, 'done':
1, "one's": 1, 'Evil': 2, 'defiled': 1}

======================================================================
```

Note: punctuation has been cleaned off the keys, i.e., no key has a period, comma, or semicolon. Upper case, plurality, and tense stay.