- **Developer**: Swathi Danturi
- **Collaborator(s)**: None
- **File Name**: Design.md
- **Date**: 09/30/2024

size() method design

To find out how many elements in the `self.notes` list, we use the Python built-in function `len()`.

- Call `len()` with argument `self.notes` and return the result.

how_often_word() method design

- `sentence` is a parameter to this method and is a string of words separated by white spaces
- Define an empty dictionary `word_occurences`
- for each `word` in `sentence.split()`, iterate through the following:
    - get the count (value) of `word` (key) in dictionary `word_occurences` using `.get()` method by giving `word` as key
    - increment the occurence count by `1`
    - assign the above value to the value of key `word` of dictionary `word_occurences[]` using assignment operator
- return `word_occurences`, which contains the key:value pairs of word:occurences

word_histogram() method design

- `self` is the current instance of the class `Notes`
- `self.notes` is an instance variable of the class `Notes` and is initialized to a list in the constructor, which is passed as an argument when the class instance is created
- call `how_often_word()` using class name instead of class instance as this is a static method
- pass the resulted string from `.join()` of `self.notes` as parameter to `how_often_words()`
- return the return value of `how_often_word()`