Homework 2

Instructions. The procedure is the same as for Homework 1. Write and submit a Python file called `hw2.py`. Indicate your answers by setting variables. For this homework, the individual questions do not say "set such-and-such a variable to the result." Instead, you should use the variable `A1` for question 1, `A2` for question 2, and so on.

1. Set `x` to 1812, then compute 16 times the sum of `x` and 120 (and set `A1` to the result).

2. Get the fourth token of `text1`. (Fourth counting from 1 in the usual way, e.g., `"d"` in the list `["a", "b", "c", "d"]`. Note: a text behaves just like a list containing strings, each string representing one word token. Also, you will of course need an `import` statement to get access to `text1`.)

3. Get the last token of `text1`.

4. Concatenate the strings `"sing"` and `"er"`.

5. Concatenate the string `"la"` with itself three times. (I.e., the result should be `"lalala"`.) The literal `"la"` should appear in your expression only once.

6. Define `words` to be a list whose elements are `'dog'`, `'cat'`, `'fish'`, and `'diplodocus'`. Concatenate its second and third elements (counting from 1).

7. Determine whether or not the fourth word in `words` (counting from 1) is at least 7 letters long. (The result will be either `True` or `False`.)

8. Determine whether the third word in `words` (counting from 1) is longer than the second word.

9. Determine whether the first letter of the first word in `words` is the same as the first letter of the last word.

10. Get the sublist consisting of just the second and third words of `words` (counting from 1). Use a slice.

11. Let `s` be the string `'garnish'`. Create a list whose first element is the first letter of `s`, whose second element is the next-to-last letter of `s`, and whose third element is the length of `s`.

12. Let the value of `w` be the string `"serendipity"`. Write an expression whose value is `"tyre"`. Your expression should contain the variable `w`, but it should not contain any string literals. Use slices.

13. Determine whether `'monstrous'` occurs more frequently in *Moby Dick* (`text1`) than it does in *Sense and Sensibility* (`text2`). Write an expression that compares the counts and returns `True` or `False`.

14. Turn the sentence `'this is a test'` into a list of words.

15. Turn the spelled word `['f', 'r', 'o', 'd', 'o']` into a string.

16. Take the sequence of 10 tokens in *Moby Dick* beginning at position 1000 (counting from 0), and turn them into a string, with words separated by spaces.

17. Determine whether or not *Moby Dick* contains the word `'vestibule'`. The result should be `True` or `False`.

18. Set `s1 = 'this is a test'` and set `s2 = 'here is another test'`. Determine which words `s1` and `s2` have in common. The result should be a set.

19. Determine how many word types are found in *Sense and Sensibility* but *not* in *Moby Dick.* The result is the number of word types.

Study

Be sure you are familiar with the constructs on Handout 2. In particular, be sure you have the following at your fingertips:

The assignment operator	`=`
The math operators	`+ - * /`
Using parentheses for grouping	
The comparison operators	`< > <= >= == !=`
The boolean values	`True False`
The logical operators	`and or not`
String literals	`'hi' "a b"`
List creation using square brackets	`[1, 2, 3]`
Getting elements of strings and lists	`s[2]`
Getting substrings and sublists	`s[2:5]`
The concatenation operator	`+`
The function `len`	
The function `print`	
The methods `split` and `join`	`s.split() ' '.join(lst)`
Set creation using braces	`{1, 2, 3}`
Set operators	`& | -`

Without using the python interpreter, you should be able to write expressions to accomplish simple tasks (like "get the number of elements in set `A`"), and you should be able to determine the value if you are given a simple expression (like `len({2,4,6})`). There will be a quiz.