

Lab 4.02 - Getting Loopy

Part 1

Write a function `pluralize_words` that takes in a list of words and updates the values of the list to make each one plural. It returns nothing.

Making plurals in English has a number of special cases, but for this lab we'll use a simple rule: if the word ends in a `y` remove the `y` and add `ies`; otherwise add an `'s'`.

We'll exercise the function on a list of words.

1. Create the function contract for `pluralize_words`.
2. Provide a few examples that confirm `pluralize_words` works as expected:
 - Include examples with 'berry'
 - What if the list is empty?

Example 1

```
# contract goes here
def pluralize_words(word_list):
    # your code goes here

word_list = ['apple', 'berry', 'melon']
print("Singular words: " + str(word_list))
pluralize_words(word_list)
print("No longer singular words: " + str(word_list))
# more examples go here
```

Here is what it should look like when you run your code

Singular words: ['apple', 'berry', 'melon']

No longer singular words: ['apples', 'berries', 'melons']

Hint

Remember that you can index into the string and get the length of a string. Use that to get the last letter of each word.

Part 2

Create a function `my_reverse`, which will return a reversed string.

1. Create the function contract for `my_reverse`.
2. Provide a few examples to confirm that `my_reverse` works:
 - An empty string
 - A string of even length
 - A string of odd length greater than 1
 - A string of length 1

Example 2

```
# contract goes here
def my_reverse(string_to_reverse):
    # your code goes here

    reversed = my_reverse("apples")
    print(reversed)
# examples go here
```

Here is what example 2 should look like when you run your code

```
>>> python3 my_reverse_lab.py
selppa
```

Hint 2

To get the last element: $(\text{len}(\text{my_list}) - 1) - 0$ To get the second to last element:
 $(\text{len}(\text{my_list}) - 1) - 1$ To get the third to last element: $(\text{len}(\text{my_list}) - 1) - 2$

Bonus

Create a function **reverse_strings_in_list**. This function will input a list of strings you want to reverse. The function will reverse the strings in the list by calling the **my_reverse** function in a loop.