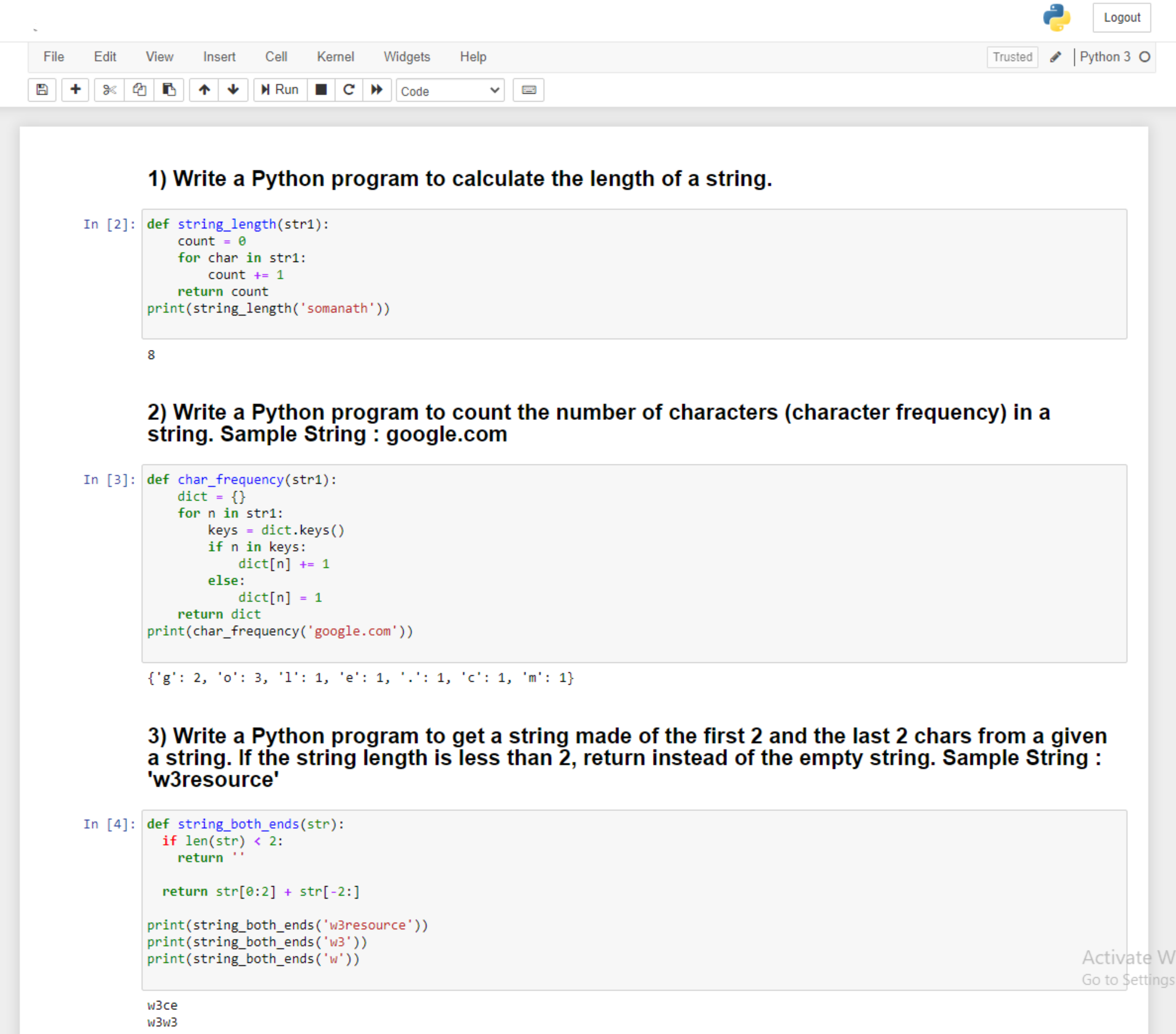


1) Write a Python program to calculate the length of a string.

```
In [2]: def string_length(str1):
            count = 0
            for char in str1:
                count += 1
            return count
        print(string_length('somanath'))
```

```
8
```

2) Write a Python program to count the number of characters (character frequency) in a string. Sample String : google.com

```
In [3]: def char_frequency(str1):
            dict = {}
            for n in str1:
                keys = dict.keys()
                if n in keys:
                    dict[n] += 1
                else:
                    dict[n] = 1
            return dict
        print(char_frequency('google.com'))
```

```
{'g': 2, 'o': 3, 'l': 1, 'e': 1, '.': 1, 'c': 1, 'm': 1}
```

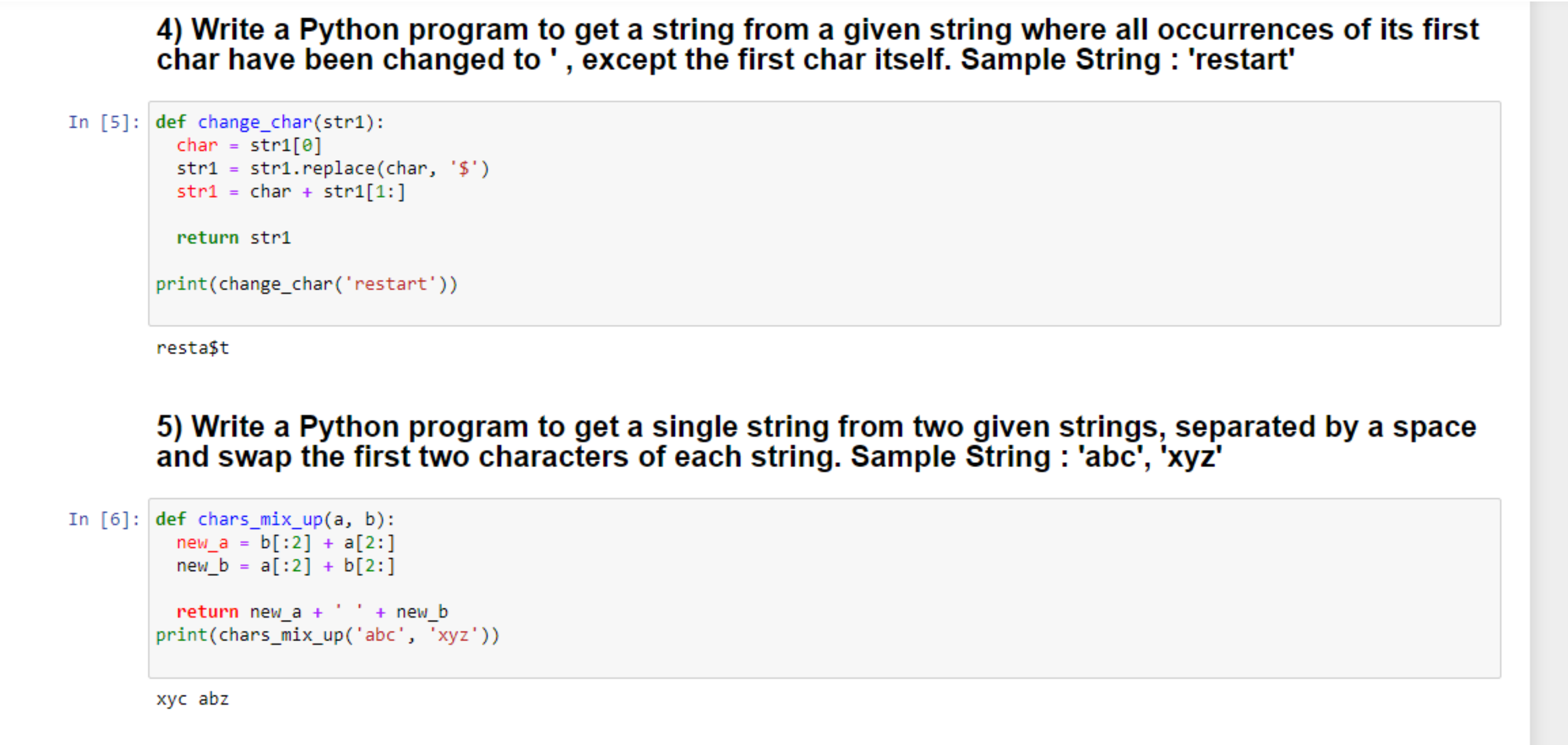
3) Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string. Sample String : 'w3resource'

```
In [4]: def string_both_ends(str):
          if len(str) < 2:
            return ''

          return str[0:2] + str[-2:]

        print(string_both_ends('w3resource'))
        print(string_both_ends('w3'))
        print(string_both_ends('w'))
```

```
w3ce
w3w3
```



4) Write a Python program to get a string from a given string where all occurrences of its first char have been changed to ' , except the first char itself. Sample String : 'restart'

In [5]:

```
def change_char(str1):
  char = str1[0]
  str1 = str1.replace(char, '$')
  str1 = char + str1[1:]

  return str1

print(change_char('restart'))
```

```
resta$t
```

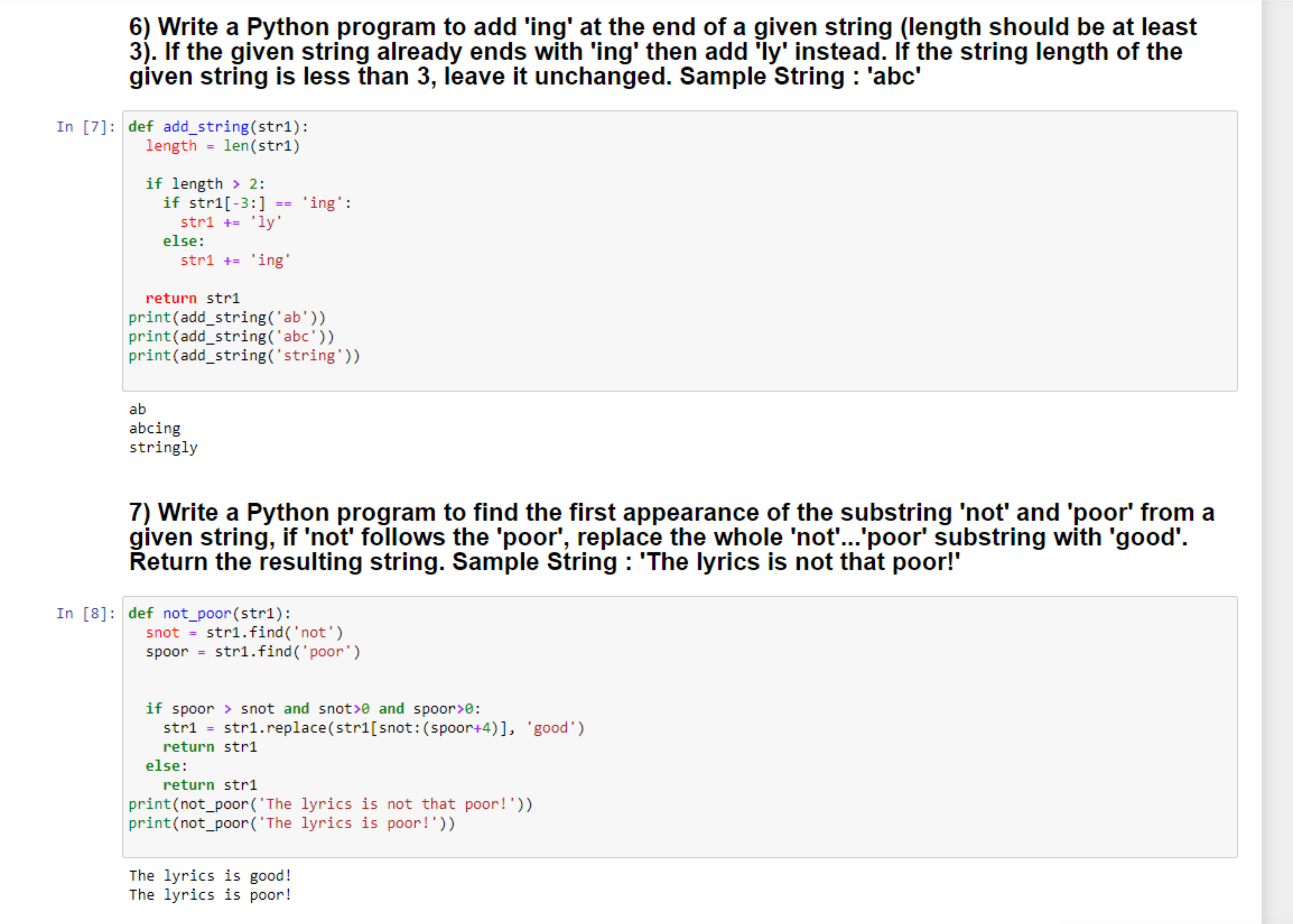
5) Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string. Sample String : 'abc', 'xyz'

In [6]:

```
def chars_mix_up(a, b):
  new_a = b[:2] + a[2:]
  new_b = a[:2] + b[2:]

  return new_a + ' ' + new_b
print(chars_mix_up('abc', 'xyz'))
```

```
xyc abz
```



6) Write a Python program to add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged. Sample String : 'abc'

```
In [7]: def add_string(str1):
          length = len(str1)

          if length > 2:
            if str1[-3:] == 'ing':
              str1 += 'ly'
            else:
              str1 += 'ing'

          return str1
        print(add_string('ab'))
        print(add_string('abc'))
        print(add_string('string'))
```

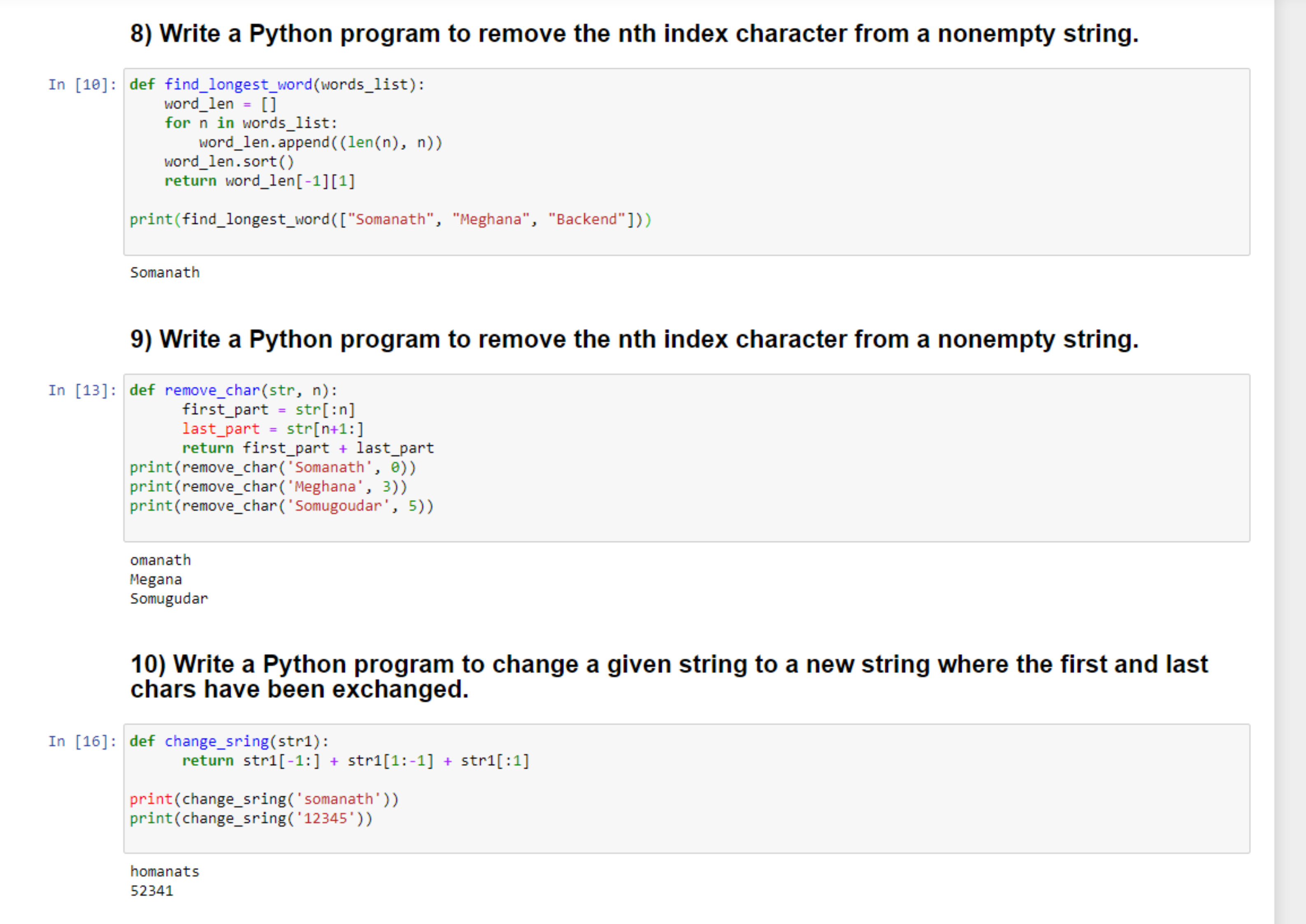
```
ab
abcing
stringly
```

7) Write a Python program to find the first appearance of the substring 'not' and 'poor' from a given string, if 'not' follows the 'poor', replace the whole 'not'...'poor' substring with 'good'. Return the resulting string. Sample String : 'The lyrics is not that poor!'

```
In [8]: def not_poor(str1):
          snot = str1.find('not')
          spoor = str1.find('poor')


          if spoor > snot and snot>0 and spoor>0:
            str1 = str1.replace(str1[snot:(spoor+4)], 'good')
            return str1
          else:
            return str1
        print(not_poor('The lyrics is not that poor!'))
        print(not_poor('The lyrics is poor!'))
```

```
The lyrics is good!
The lyrics is poor!
```



8) Write a Python program to remove the nth index character from a nonempty string.

In [10]:

```
def find_longest_word(words_list):
    word_len = []
    for n in words_list:
        word_len.append((len(n), n))
    word_len.sort()
    return word_len[-1][1]

print(find_longest_word(["Somanath", "Meghana", "Backend"]))
```

```
Somanath
```

9) Write a Python program to remove the nth index character from a nonempty string.

In [13]:

```
def remove_char(str, n):
      first_part = str[:n]
      last_part = str[n+1:]
      return first_part + last_part
print(remove_char('Somanath', 0))
print(remove_char('Meghana', 3))
print(remove_char('Somugoudar', 5))
```

```
omanath
Megana
Somugudar
```

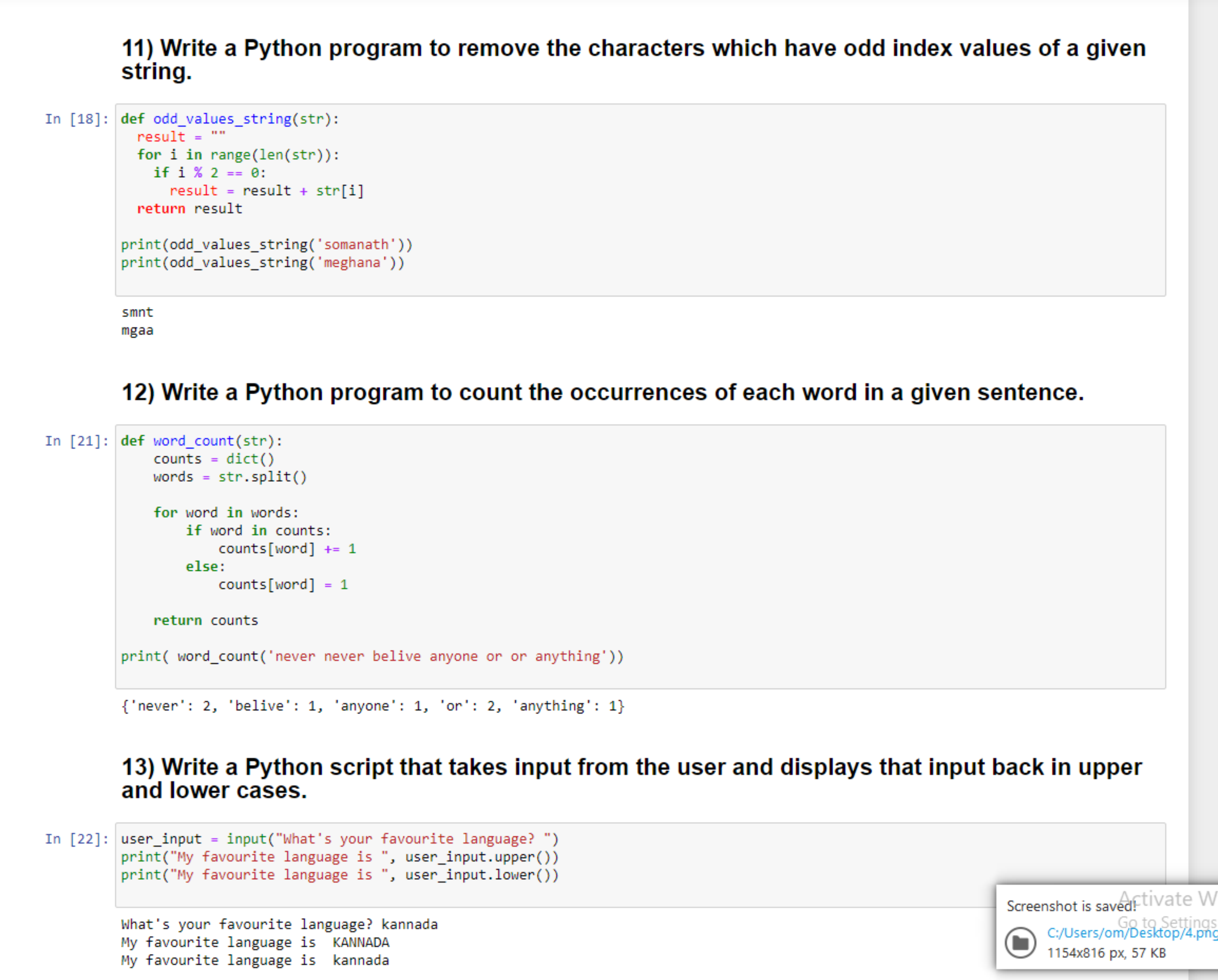
10) Write a Python program to change a given string to a new string where the first and last chars have been exchanged.

In [16]:

```
def change_sring(str1):
      return str1[-1:] + str1[1:-1] + str1[:1]

print(change_sring('somanath'))
print(change_sring('12345'))
```

```
homanats
52341
```



11) Write a Python program to remove the characters which have odd index values of a given string.

In [18]:
```
def odd_values_string(str):
  result = ""
  for i in range(len(str)):
    if i % 2 == 0:
      result = result + str[i]
  return result

print(odd_values_string('somanath'))
print(odd_values_string('meghana'))
```

```
smnt
mgaa
```

12) Write a Python program to count the occurrences of each word in a given sentence.

In [21]:
```
def word_count(str):
    counts = dict()
    words = str.split()

    for word in words:
        if word in counts:
            counts[word] += 1
        else:
            counts[word] = 1

    return counts

print( word_count('never never belive anyone or or anything'))
```

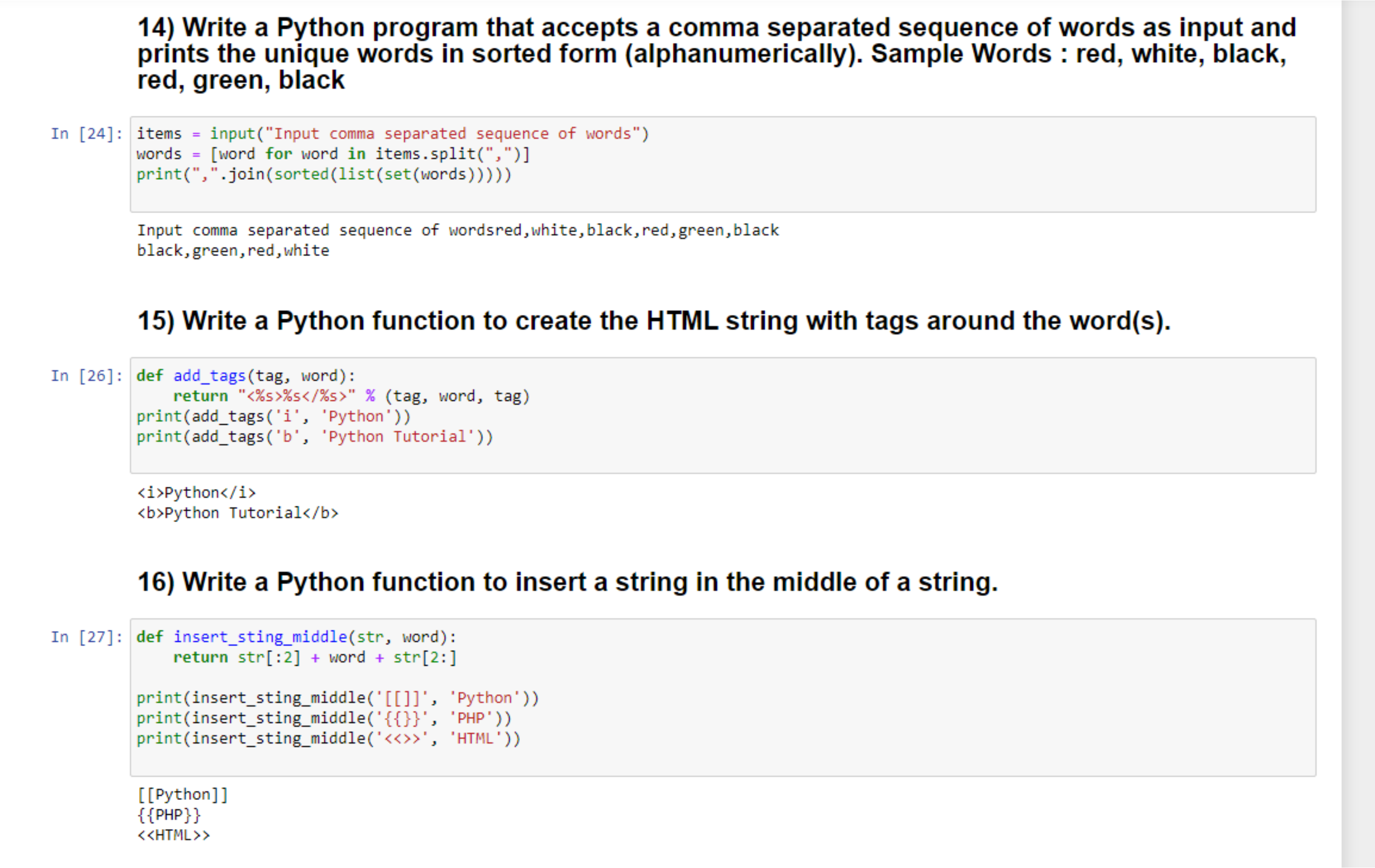
```
{'never': 2, 'belive': 1, 'anyone': 1, 'or': 2, 'anything': 1}
```

13) Write a Python script that takes input from the user and displays that input back in upper and lower cases.

In [22]:
```
user_input = input("What's your favourite language? ")
print("My favourite language is ", user_input.upper())
print("My favourite language is ", user_input.lower())
```

```
What's your favourite language? kannada
My favourite language is  KANNADA
My favourite language is  kannada
```

Screenshot is saved!
C:/Users/om/Desktop/4.png
1154x816 px, 57 KB



14) Write a Python program that accepts a comma separated sequence of words as input and prints the unique words in sorted form (alphanumerically). Sample Words : red, white, black, red, green, black

```
In [24]: items = input("Input comma separated sequence of words")
         words = [word for word in items.split(",")]
         print(",".join(sorted(list(set(words)))))
```

```
Input comma separated sequence of wordsred,white,black,red,green,black
black,green,red,white
```

15) Write a Python function to create the HTML string with tags around the word(s).

```
In [26]: def add_tags(tag, word):
             return "<%s>%s</%s>" % (tag, word, tag)
         print(add_tags('i', 'Python'))
         print(add_tags('b', 'Python Tutorial'))
```

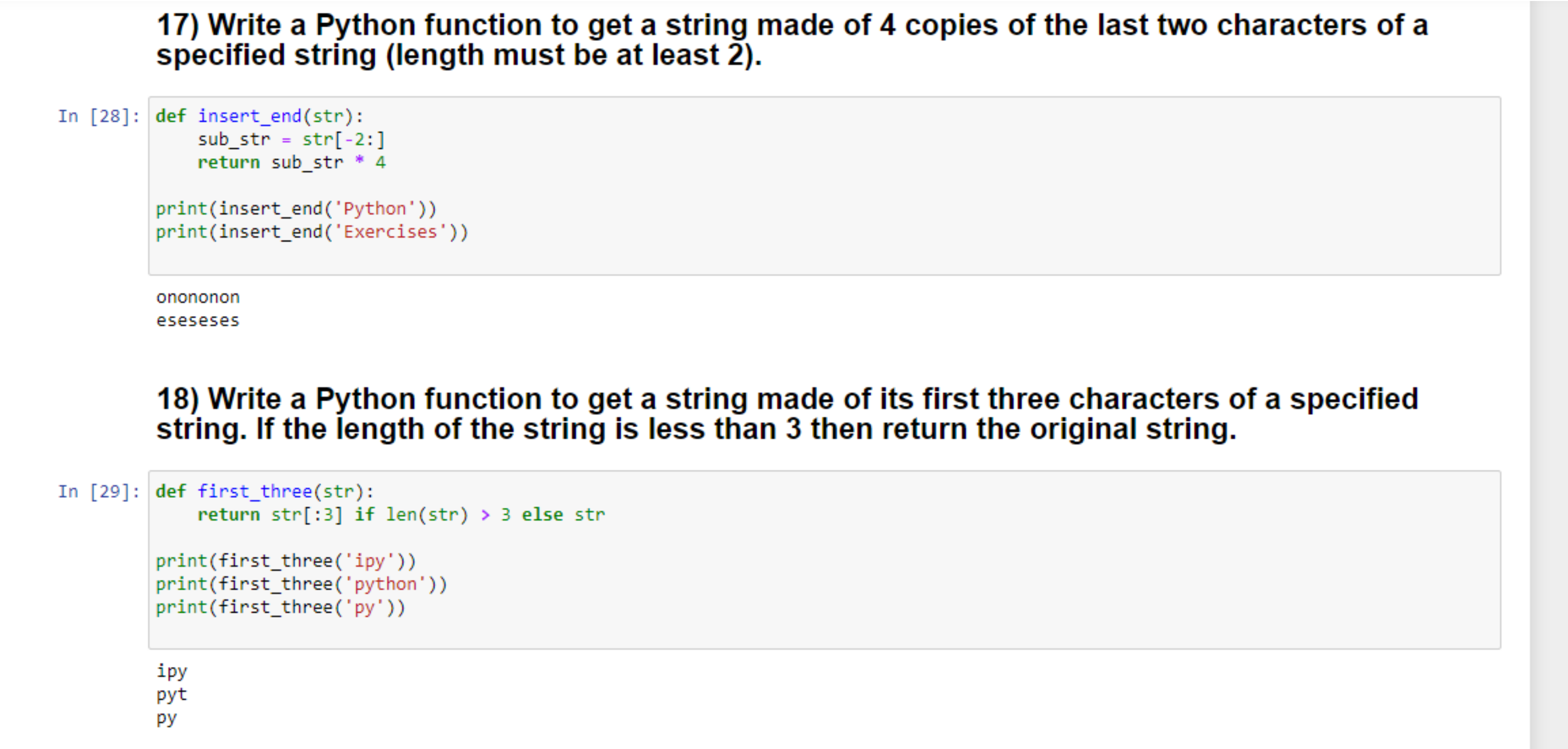
```
<i>Python</i>
<b>Python Tutorial</b>
```

16) Write a Python function to insert a string in the middle of a string.

```
In [27]: def insert_sting_middle(str, word):
             return str[:2] + word + str[2:]

         print(insert_sting_middle('[[]]', 'Python'))
         print(insert_sting_middle('{{}}', 'PHP'))
         print(insert_sting_middle('<<>>', 'HTML'))
```

```
[[Python]]
{{PHP}}
<<HTML>>
```



17) Write a Python function to get a string made of 4 copies of the last two characters of a specified string (length must be at least 2).

In [28]:

```
def insert_end(str):
    sub_str = str[-2:]
    return sub_str * 4

print(insert_end('Python'))
print(insert_end('Exercises'))
```

```
onononon
eseseses
```

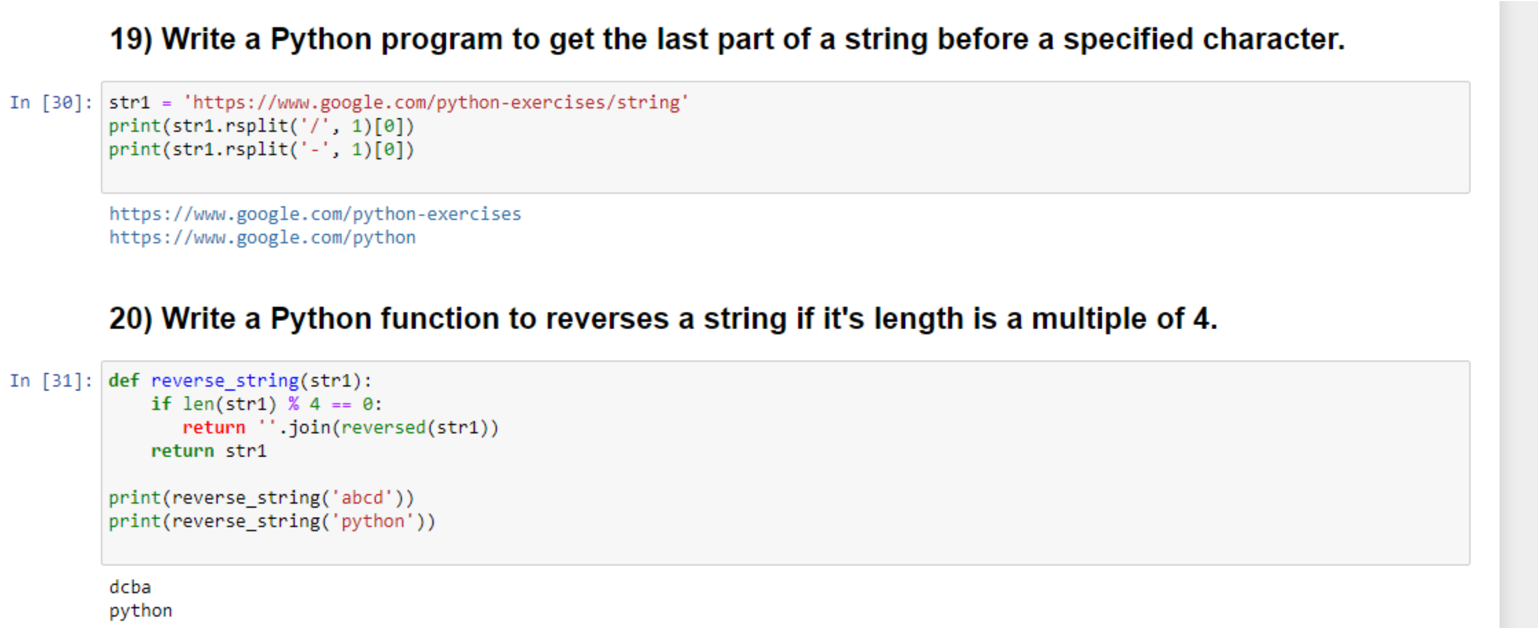
18) Write a Python function to get a string made of its first three characters of a specified string. If the length of the string is less than 3 then return the original string.

In [29]:

```
def first_three(str):
    return str[:3] if len(str) > 3 else str

print(first_three('ipy'))
print(first_three('python'))
print(first_three('py'))
```

```
ipy
pyt
py
```



19) Write a Python program to get the last part of a string before a specified character.

In [30]:

```python
str1 = 'https://www.google.com/python-exercises/string'
print(str1.rsplit('/', 1)[0])
print(str1.rsplit('-', 1)[0])
```

```
https://www.google.com/python-exercises
https://www.google.com/python
```

20) Write a Python function to reverses a string if it's length is a multiple of 4.

In [31]:

```python
def reverse_string(str1):
    if len(str1) % 4 == 0:
       return ''.join(reversed(str1))
    return str1

print(reverse_string('abcd'))
print(reverse_string('python'))
```

```
dcba
python
```