

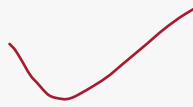
misc_exercise

10/10

May 28, 2024

1) Reverse a string

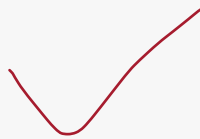
```
[ ]: def reverse_string(string):  
      reversed = string[::-1]  
      return reversed  
  
reverse_string("market")
```



```
[ ]: 'tekram'
```

2) Palindrome Checker

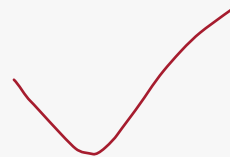
```
[ ]: def palindrome_checker(string):  
      rev_string = string[::-1]  
      if rev_string == string:  
          return True  
      else:  
          return False  
  
palindrome_checker("ant")
```



```
[ ]: False
```

3) Factorial Calculator

```
[ ]: def factorial(num):  
      if num.is_integer():  
          if num > 0:  
              ans = num * factorial(num-1)  
              return ans  
          elif num == 0:  
              ans = 1  
              return ans  
          else:  
              print("Enter positive integer")  
      else:  
          print("Enter positive integer")
```



```
factorial(5)
```

```
[ ]: 120
```

4) Fibonacci Sequence

```
[ ]: #Fn = Fn-1 + Fn-2, with 0 and 1 being the first two numbers  
#to generate sequence, we use for loop:  
def fibonacci(num):  
    #the first two numbers are 0 and 1:  
    series = [0,1]  
    for i in range(2, num):  
        series.append(series[-1] + series[-2])  
    return series  
  
fibonacci(8)
```

```
[ ]: [0, 1, 1, 2, 3, 5, 8, 13]
```

```
[ ]: series = [0,1]  
     series[-1]
```

```
[ ]: 1
```

```
[ ]: print(range(7))
```

```
range(0, 7)
```

5) Prime number checker

```
[ ]: #to check for prime number, the divisors that would result in 0 reminders, ␣  
↪ aside from 1 lie between 2 and number/2  
#so loop for all numbers in that range:  
  
def prime_num(number):  
    if number > 1:  
        for i in range(2, (number//2)+1):  
            if (number%i) == 0:  
                print("Not a prime number")  
                break  
            else:  
                print("This a prime number")  
    else:  
        print("Not a prime number")  
  
prime_num(5)
```

```
This a prime number
```

6) List reversal

```
[ ]: def reverse_list(list_of_things):  
    list_of_things.reverse()  
    return list_of_things  
  
reverse_list([1,2,3,4,5])
```

```
[ ]: [5, 4, 3, 2, 1]
```

7) List sorting

```
[ ]: #don't really understand this one:  
def sorting(mylist):  
    for i in range(0, len(mylist)):  
        for j in range(i+1, len(mylist)):  
            if mylist[i] >= mylist[j]:  
                temp = mylist[i]  
                mylist[i] = mylist[j]  
                mylist[j] = temp  
    return mylist  
  
print(sorting([78,23,98,54,62,13,65]))
```

```
[13, 23, 54, 62, 65, 78, 98]
```

8) Anagram checker

```
[ ]: def anagram(string1, string2):  
    #empty dictionaries to store results:  
    f1 = {}  
    f2 = {}  
  
    #valid letters (to eliminate other characters from being counted):  
    valid_letters = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n',  
                    'o','p','q','r','s','t','u','v','w','x','y','z','A','B','C','D','E','F','G','H','I','J','K',  
                    'O','P','Q','R','S','T','U','V','W','X','Y','Z']  
  
    for char in set(string1):  
        if char in valid_letters:  
            #dict[key] = value  
            f1[char] = string1.count(char)  
  
    for char in set(string2):  
        if char in valid_letters:  
            #dict[key] = value  
            f2[char] = string2.count(char)
```

```
#check if dictionaries are equal:
if f1 == f2:
    return True
else:
    return False

anagram("annaapple", "apple anna")
```

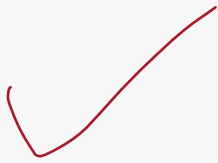
[]: True

9) Count words in string

```
[ ]: #words are separated by spaces:

def word_count(string):
    words = string.split(" ")
    return len(words)

word_count("I love maths and music.")
```

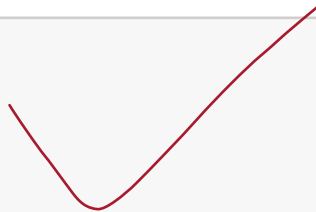


[]: 5

10) Unique elements

```
[ ]: #use set() to remove duplicates
def unique_elements(original_list):
    unique_contents = set(original_list)
    return list(unique_contents)

unique_elements([1,2,3,4,2,1,4,5,2,2,4])
```



[]: [1, 2, 3, 4, 5]