Possible solutions to the challenge

Challenge 1 tuple

def string\_fun(string):  
Return a tuple with three elements  
The first element is True if the string contains only alphabet characters, otherwise False  
The second element is True if the string ends with an exclamation mark('!'), otherwise False  
The third element is the string with all spaces (' ') replaced with hyphens ('-')

#code9: test tuple

tup2 = ("cloudacademy",)

details = ("hello", "world!", "string\_with\_hyphens")

print(f"tup\_test = {details}")

print()

Gives the output:

tup\_test = ('hello', 'world!', 'string\_with\_hyphens')

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Other person solution

Challenge Two:

def list\_uniqueness(the\_list):  
Return a dictionary with two key-value pairs:

1. The key 'list\_length' stores the length of the\_list as its value

2. The key 'unique\_items' stores the number of unique items in the\_list as its value

#CODE3: Create DICTIONARY with multiple key-value pairs

print("Test\_Challenge:")

dict3 = {"name": "cloudacademy", "count": 1000}

print(f"length = {len(dict3)}")

print(f"values = {d.itervalues(dict3)}")

print()

#print("CODE3:")

#print(f"dict3 = {dict3}")

#print(f"data type = {type(dict3)}")

Challenge Two:  
dictionary = {"list\_length": "the\_list", "unique\_items": "the\_list"}  
print(f"dictionary = {dictionary})  
print()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Challenge Three:

def append(the\_list, item):  
Append item to the\_list  
Layout:  
new\_list = the\_list  
new\_list = new\_list + [item]  
the\_list = new\_list