

[SOLUTION] Assignment 01

March 7, 2023

0.0.1 Problem 01: Reverse a List

Write a function that takes a list of integers as input and returns the list in reverse order.

Examples:

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

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

Input: [6, 7, 8, 9, 10]

Output: [10, 9, 8, 7, 6]

```
[1]: def reverse_list(li):  
      return li[::-1]  
  
      print(reverse_list([1, 2, 3, 4, 5]))  
      print(reverse_list([6, 7, 8, 9, 10]))
```

[5, 4, 3, 2, 1]

[10, 9, 8, 7, 6]

0.0.2 Problem 02: Convert Celsius to Fahrenheit

Write a function that takes a temperature in Celsius as input and returns the equivalent temperature in Fahrenheit.

The formula for converting Celsius to Fahrenheit is:

$$F = (C * 9/5) + 32$$

Examples:

Input: 0

Output: 32

Input: 37

Output: 98.6

```
[2]: def celsius_to_fahrenheit(celsius):  
      fahrenheit = (celsius * 9/5) + 32  
      return fahrenheit
```

```
print(celsius_to_fahrenheit(0))
print(celsius_to_fahrenheit(37))
```

32.0
98.6

0.0.3 Problem 03: Count the Occurrences of Each Word

Write a function that takes a string as input and returns a dictionary where the keys are the words in the string and the values are the number of times each word appears.

Examples:

Input: "the cat in the hat"

Output: {'the': 2, 'cat': 1, 'in': 1, 'hat': 1}

Input: "this is a test this is a test"

Output: {'this': 2, 'is': 2, 'a': 2, 'test': 2}

```
[3]: def count_words(string):
      words = string.split()
      word_count = {}
      for word in words:
          if word in word_count:
              word_count[word] += 1
          else:
              word_count[word] = 1
      return word_count

      print(count_words("the cat in the hat"))
      print(count_words("this is a test this is a test"))
```

{'the': 2, 'cat': 1, 'in': 1, 'hat': 1}
{'this': 2, 'is': 2, 'a': 2, 'test': 2}

0.0.4 Problem 04: Check if a Number is Prime

Write a function that takes a positive integer as input and returns True if the number is prime, and False if it is not.

Examples:

Input: 2

Output: True

Input: 8

Output: False

```
[4]: def is_prime(num):
      if num <= 1:
          return False
```

```

    for i in range(2, num):
        if num % i == 0:
            return False
    return True

print(is_prime(2))
print(is_prime(8))

```

True
False

0.0.5 Problem 05: Find the Longest Word

Write a function that takes a list of strings as input and returns the longest word in the list.

Examples:

Input: ["hello", "world", "this", "is", "a", "test"]

Output: "hello"

Input: ["how", "are", "you", "today"]

Output: "today"

```

[5]: def longest_word(words):
    longest = ""
    for word in words:
        if len(word) > len(longest):
            longest = word
    return longest

print(longest_word(["hello", "world", "this", "is", "a", "test"]))
print(longest_word(["how", "are", "you", "today"]))

```

hello
today

0.0.6 Problem 06: Calculate the Sum of a List of Numbers

Write a function that takes a list of numbers as input and returns the sum of all the numbers in the list.

Examples:

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

Output: 15

Input: [6, 7, 8, 9, 10]

Output: 40

```

[6]: def sum_of_numbers(numbers):
    total = 0

```

```

    for num in numbers:
        total += num
    return total

print(sum_of_numbers([1, 2, 3, 4, 5]))
print(sum_of_numbers([6, 7, 8, 9, 10]))

```

15
40

0.0.7 Problem 07: Check if a Number is Even or Odd

Write a function that takes a positive integer as input and returns “Even” if the number is even, and “Odd” if it is odd.

Examples:

Input: 2
Output: "Even"

Input: 5
Output: "Odd"

```

[7]: def even_or_odd(num):
    if num % 2 == 0:
        return "Even"
    else:
        return "Odd"

print(even_or_odd(2))
print(even_or_odd(5))

```

Even
Odd

0.0.8 Problem 08: Get the Factorial of a Number

Write a function that takes a positive integer as input and returns the factorial of the number.

Examples:

Input: 5
Output: 120

Input: 7
Output: 5040

```

[8]: def factorial(num):
    result = 1
    for i in range(1, num+1):
        result *= i

```

```
    return result

print(factorial(5))
print(factorial(7))
```

120
5040

0.0.9 Problem 09: Convert a List to a Tuple

Write a function that takes a list as input and returns a tuple with the same elements as the list.

Examples:

Input: [1, 2, 3, 4, 5]
Output: (1, 2, 3, 4, 5)

Input: ["apple", "banana", "cherry"]
Output: ("apple", "banana", "cherry")

```
[9]: def list_to_tuple(lst):
      return tuple(lst)

      print(list_to_tuple([1, 2, 3, 4, 5]))
      print(list_to_tuple(["apple", "banana", "cherry"]))
```

(1, 2, 3, 4, 5)
(*'apple'*, *'banana'*, *'cherry'*)

0.0.10 Problem 10: Check if an Element is in a Tuple

Write a function that takes a tuple and an element as input and returns True if the element is in the tuple, and False if it is not.

Examples:

Input: (1, 2, 3, 4, 5), 3
Output: True

Input: (6, 7, 8, 9, 10), 11
Output: False

```
[10]: def is_element_in_tuple(tpl, element):
      return element in tpl

      print(is_element_in_tuple((1, 2, 3, 4, 5), 3))
      print(is_element_in_tuple((6, 7, 8, 9, 10), 11))
```

True
False

0.0.11 Problem 11: Merge Two Dictionaries

Write a function that takes two dictionaries as input and returns a new dictionary that contains the key-value pairs of both dictionaries. In case of overlapping keys, the values of the second dictionary should overwrite the values of the first dictionary.

Examples:

Input: {1: "one", 2: "two"}, {2: "two", 3: "three"}

Output: {1: "one", 2: "two", 3: "three"}

Input: {"a": 1, "b": 2}, {"c": 3, "d": 4}

Output: {"a": 1, "b": 2, "c": 3, "d": 4}

```
[11]: def merge_dicts(dict1, dict2):  
        result = dict1.copy()  
        result.update(dict2)  
        return result  
  
print(merge_dicts({1: "one", 2: "two"}, {2: "two", 3: "three"}))  
print(merge_dicts({"a": 1, "b": 2}, {"c": 3, "d": 4}))
```

{1: 'one', 2: 'two', 3: 'three'}

{'a': 1, 'b': 2, 'c': 3, 'd': 4}

0.0.12 Problem 12: Check if Two Sets Have Any Common Elements

Write a function that takes two sets as input and returns True if the sets have any common elements, and False if they do not.

Examples:

Input: {1, 2, 3, 4, 5}, {4, 5, 6, 7, 8}

Output: True

Input: {6, 7, 8, 9, 10}, {11, 12, 13, 14, 15}

Output: False

```
[12]: def common_elements(set1, set2):  
        return len(set1 & set2) > 0  
  
print(common_elements({1, 2, 3, 4, 5}, {4, 5, 6, 7, 8}))  
print(common_elements({6, 7, 8, 9, 10}, {11, 12, 13, 14, 15}))
```

True

False

0.0.13 Problem 13: Reverse a String

Write a function that takes a string as input and returns the string in reverse order.

Examples:

Input: "hello"
Output: "olleh"

Input: "world"
Output: "dlrow"

```
[13]: def reverse_string(string):  
        return string[::-1]  
  
print(reverse_string("hello"))  
print(reverse_string("world"))
```

olleh
dlrow

0.0.14 Problem 14: Count the Occurrences of a Substring

Write a function that takes a string and a substring as input and returns the number of occurrences of the substring in the string.

Examples:

Input: "hello world", "o"
Output: 2

Input: "this is a test", "i"
Output: 2

```
[14]: def count_substring(string, sub):  
        return string.count(sub)  
  
print(count_substring("hello world", "o"))  
print(count_substring("this is a test", "i"))
```

2
2

0.0.15 Problem 15: Create a Rectangle Class

Create a class called Rectangle that has two attributes: length and width. The class should have a method called get_area that returns the area of the rectangle and a method called get_perimeter that returns the perimeter of the rectangle.

Examples:

```
rect = Rectangle(5, 10)  
print(rect.get_length()) # 5  
print(rect.get_width()) # 10  
print(rect.get_area()) # 50  
print(rect.get_perimeter()) # 30
```

```
[15]: class Rectangle:
    def __init__(self, length = 0, width = 0):
        self.__length = length
        self.__width = width

    def get_length(self):
        return self.__length

    def get_width(self):
        return self.__width

    def get_area(self):
        return self.__length * self.__width

    def get_perimeter(self):
        return 2 * (self.__length + self.__width)
```

```
[16]: rect = Rectangle(5, 10)
print(rect.get_length()) # 5
print(rect.get_width()) # 10
print(rect.get_area()) # 50
print(rect.get_perimeter()) # 30
```

```
5
10
50
30
```

0.0.16 Problem 16: Create a Car Class

Create a class called Car that has two attributes: make and model. The class should have a method called info that returns the make and model of the car. The class should also have a method called drive that returns the message “Driving [make] [model]” where [make] and [model] are the make and model of the car.

Examples:

```
car = Car("Toyota", "Camry")
print(car.get_make) # "Toyota"
print(car.get_model) # "Camry"
print(car.get_info()) # "Toyota Camry"
print(car.drive()) # "Driving Toyota Camry"
```

```
[17]: class Car:
    def __init__(self, make = "", model = ""):
        self.__make = make
        self.__model = model

    def get_make(self):
```



```

        return self.__make

    def get_model(self):
        return self.__model

    def get_info(self):
        return f"{self.__make} {self.__model}"

    def drive(self):
        return f"Driving {self.__make} {self.__model}"

```

```

[18]: car = Car("Toyota", "Camry")
print(car.get_make()) # "Toyota"
print(car.get_model()) # "Camry"
print(car.get_info()) # "Toyota Camry"
print(car.drive()) # "Driving Toyota Camry"

```

```

Toyota
Camry
Toyota Camry
Driving Toyota Camry

```

0.0.17 Problem 17: Read a CSV File

Write a function that takes a CSV file as input and returns a list of dictionaries, where each dictionary represents a row in the CSV file. The keys in each dictionary should be the header row of the CSV file and the values should be the corresponding values for each row.

Examples:

CSV file: (create file with given data if needed)

```

Name, Age, Gender
John, 30, Male
Jane, 25, Female

```

Output:

```

[{"Name": "John", "Age": "30", "Gender": "Male"}, {"Name": "Jane", "Age": "25", "Gender": "Female"}]

```

```

[19]: import csv

def read_csv(file_path):
    with open(file_path, "r") as file:
        reader = csv.DictReader(file)
        return list(reader)

print(read_csv("data.csv"))

```

```

[{'Name': 'John', 'Age': '30', 'Gender': 'Male'}, {'Name': 'Jane', 'Age': '25', 'Gender': 'Female'}, {'Name': 'John', 'Age': '30', 'Gender': 'Male'}]

```

0.0.18 Problem 18: Write a CSV File

Write a function that takes a list of dictionaries and a CSV file path as input and writes the dictionaries to the CSV file, creating a new file or overwriting an existing file. The keys of each dictionary should be used as the header row in the CSV file.

Examples:

Input:

```
data = [{"Name": "John", "Age": "30", "Gender": "Male"},
        {"Name": "Jane", "Age": "25", "Gender": "Female"}]
file_path = "data.csv"
```

Output:

```
CSV file:
Name, Age, Gender
John, 30, Male
Jane, 25, Female
```

```
[20]: import csv

def write_csv(data, file_path):
    with open(file_path, "w", newline='') as file:
        writer = csv.DictWriter(file, fieldnames=data[0].keys())
        writer.writeheader()
        for row in data:
            writer.writerow(row)

data = [{"Name": "John", "Age": "30", "Gender": "Male"},
        {"Name": "Jane", "Age": "25", "Gender": "Female"}]
filepath = "data.csv"
write_csv(data, filepath)
```

0.0.19 Problem 19: Divide Numbers

Write a function that takes two numbers as input and returns their division. If the second number is 0, the function should raise a `ValueError` with the message “Cannot divide by zero.”

Examples:

Input:

```
num1 = 10
num2 = 2
```

Output: 5

Input:

```
num1 = 10
num2 = 0
```

Output: ValueError: Cannot divide by zero.

```
[21]: def divide_numbers(num1, num2):  
        if num2 == 0:  
            raise ValueError("Cannot divide by zero.")  
        return num1 / num2  
  
print(divide_numbers(10, 2))  
print(divide_numbers(10, 0))
```

5.0

```
-----  
ValueError                                Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_1952\1823938946.py in <module>  
      5  
      6 print(divide_numbers(10, 2))  
----> 7 print(divide_numbers(10, 0))  
  
~\AppData\Local\Temp\ipykernel_1952\1823938946.py in divide_numbers(num1, num2)  
      1 def divide_numbers(num1, num2):  
      2     if num2 == 0:  
----> 3         raise ValueError("Cannot divide by zero.")  
      4     return num1 / num2  
      5  
  
ValueError: Cannot divide by zero.
```

0.0.20 Problem 20: Find Unique Values in a Column

Write a function that takes a CSV file and a column name as input and returns a list of the unique values in that column.

Examples:

CSV file: (create file with given data if needed)

Name, Age, Gender

John, 30, Male

Jane, 25, Female

John, 30, Male

Input:

```
file_path = "data.csv"
```

```
column_name = "Name"
```

Output: ["John", "Jane"]

```
[22]: import csv

def find_unique_values(file_path, column_name):
    with open(file_path, "r") as file:
        reader = csv.DictReader(file)
        values = [row[column_name] for row in reader]
        return list(set(values))

print(find_unique_values("data.csv", "Name"))
```

```
['John', 'Jane']
```