# Python Assignment Solutions

This document contains solutions to the Python assignment provided. Each question has been addressed with detailed explanations and code examples where necessary.

## 1. Key Features of Python

Python is known for its simplicity, readability, and extensive libraries. Some of the key features include:

- Easy to learn and use

- Interpreted language

- Cross-platform compatibility

- Extensive standard library

- Supports multiple programming paradigms

- Dynamically typed

## 2. Data Types in Python

Python supports various data types, including:

- int: Integer numbers

- float: Floating-point numbers

- str: Strings

- list: Lists

- tuple: Tuples

- dict: Dictionaries

- set: Sets

## 3. Local and Global Variables in Python

Local variables are declared inside a function and can only be used within that function. Global variables are declared outside all functions and can be accessed from any function in the program.

Example:

global\_var = "I am global"  
  
def my\_function():  
 local\_var = "I am local"  
 print(local\_var)  
 print(global\_var)  
  
my\_function()  
print(global\_var)  
# print(local\_var) # This will cause an error as local\_var is not accessible outside the function

## 4. Comments in Python

Comments are important for documenting the code. Single-line comments start with # and multi-line comments are enclosed in triple quotes.

Example:

# This is a single-line comment

"""  
This is a   
multi-line comment  
"""

## 5. Python Literals

Literals are data given in a variable or constant. Python supports several types of literals:

- String literals: "Hello", 'World'

- Numeric literals: 123, 4.56

- Boolean literals: True, False

- Special literals: None

## 6. Variable Assignments

Variables can be assigned values in various ways:

x = 5

y = "Hello"

x, y, z = 1, 2, 3

a = b = c = "same value"

## 7. Escape Characters in Python

Escape characters are used to represent certain whitespace characters:

\n: New line

\t: Tab

\': Single quote

\" : Double quote

\\b: Backspace

## 8. String Formatting in Python

Various methods to format strings include:

# Using % operator

name = "John"  
print("Hello, %s!" % name)

# Using str.format()

age = 25  
print("I am {} years old".format(age))

# Using f-strings (Python 3.6+)

print(f"My name is {name} and I am {age} years old")

## 9. Practical Tasks

### Task 10: Print Each Character of a String in a New Line

string = input("Enter a string: ")  
for char in string:  
 print(char)

### Task 11: Find the Length of the String "machine learning"

# Using len() function  
string = "machine learning"  
length = len(string)  
print(f"Length of the string: {length}")  
  
# Without using len() function  
length = 0  
for char in string:  
 length += 1  
print(f"Length of the string: {length}")

### Task 12: Check if "orange" is Present in "This is orange juice"

string = "This is orange juice"  
if "orange" in string:  
 print("The word 'orange' is present.")  
else:  
 print("The word 'orange' is not present.")

**Task 13: Find the Number of Vowels, Consonants, Digits, and Whitespace Characters in a String**  
string = input("Enter a string: ")

vowels = consonants = digits = whitespaces = 0

for char in string:

if char.lower() in 'aeiou':

vowels += 1

elif char.isdigit():

digits += 1

elif char.isspace():

whitespaces += 1

elif char.isalpha():

consonants += 1

print(f"Vowels: {vowels}, Consonants: {consonants}, Digits: {digits}, Whitespaces: {whitespaces}")

## Task 14: Count Uppercase, Lowercase, Special Characters, and Numeric Values in a Given String

string = input("Enter a string: ")

uppercase = lowercase = special = numeric = 0

for char in string:

if char.isupper():

uppercase += 1

elif char.islower():

lowercase += 1

elif char.isdigit():

numeric += 1

else:

special += 1

print(f"Uppercase: {uppercase}, Lowercase: {lowercase}, Special: {special}, Numeric: {numeric}")

### Task 15: Make a New String with All the Consonants Deleted

string = "Hello, have a good day"

vowels = "aeiouAEIOU"

new\_string = ''.join([char for char in string if char in vowels or not char.isalpha()])

print("New string with all consonants deleted:", new\_string)

### Task 16: Remove the nth Index Character from a Non-Empty String

def remove\_nth\_index(string, n):

if n < 0 or n >= len(string):

return string

return string[:n] + string[n+1:]

string = input("Enter a string: ")

n = int(input("Enter the index to remove: "))

new\_string = remove\_nth\_index(string, n)

print("String after removing the nth character:", new\_string)

### Task 17: Change a Given String to a New String Where the First and Last Characters Have Been Exchanged

def exchange\_first\_last(string):

if len(string) <= 1:

return string

return string[-1] + string[1:-1] + string[0]

string = input("Enter a string: ")

new\_string = exchange\_first\_last(string)

print("String with first and last characters exchanged:", new\_string)

### Task 18: Count the Occurrences of Each Word in a Given Sentence

from collections import Counter

sentence = input("Enter a sentence: ")

words = sentence.split()

word\_count = Counter(words)

print("Word occurrences:", word\_count)

### Task 19: Count the Occurrence of a Given Character in a String

string = input("Enter a string: ")

char = input("Enter a character to count: ")

count = string.count(char)

print(f"The character '{char}' occurs {count} times in the string.")

### Task 20: Find Last 10 Characters of a String

string = input("Enter a string: ")

last\_10\_chars = string[-10:]

print("Last 10 characters of the string:", last\_10\_chars)

### Task 21: Convert a Given String to All Uppercase if It Contains at Least 2 Uppercase Characters in the First 4 Characters

def to\_upper\_if\_condition\_met(string):

if sum(1 for char in string[:4] if char.isupper()) >= 2:

return string.upper()

return string

string = input("Enter a string: ")

new\_string = to\_upper\_if\_condition\_met(string)

print("Transformed string:", new\_string)

### Task 22: Remove a Newline in Python

string = "Hello,\nWorld!\n"

new\_string = string.replace("\n", "")

print("String with newlines removed:", new\_string)

### Task 23: Swap Commas and Dots in a String

def swap\_commas\_dots(string):

return string.replace('.', '#').replace(',', '.').replace('#', ',')

string = "32.054,23"

new\_string = swap\_commas\_dots(string)

print("String after swapping commas and dots:", new\_string)

### Task 24: Find the First Repeated Character in a Given String

def first\_repeated\_char(string):

seen = set()

for char in string:

if char in seen:

return char

seen.add(char)

return None

string = input("Enter a string: ")

repeated\_char = first\_repeated\_char(string)

if repeated\_char:

print(f"The first repeated character is '{repeated\_char}'.")

else:

print("No repeated characters found.")

### Task 25: Find the Second Most Repeated Word in a Given String

from collections import Counter

def second\_most\_repeated\_word(string):

words = string.split()

word\_count = Counter(words)

most\_common = word\_count.most\_common()

if len(most\_common) < 2:

return None

return most\_common[1][0]

string = input("Enter a string: ")

second\_most\_word = second\_most\_repeated\_word(string)

if second\_most\_word:

print(f"The second most repeated word is '{second\_most\_word}'.")

else:

print("No second most repeated word found.")

### Task 26: Count Even and Odd Numbers in a String

def count\_even\_odd(string):

even = odd = 0

for char in string:

if char.isdigit():

if int(char) % 2 == 0:

even += 1

else:

odd += 1

return even, odd

string = input("Enter a string: ")

even\_count, odd\_count = count\_even\_odd(string)

print(f"Even digits: {even\_count}, Odd digits: {odd\_count}")

### Task 27: Check if a String Contains Only Digits

string = input("Enter a string: ")

if string.isdigit():

print("The string contains only digits.")

else:

print("The string does not contain only digits.")

### Task 28: Remove a Given Character/Word from String

def remove\_character(string, char):

return string.replace(char, "")

string = input("Enter a string: ")

char = input("Enter the character/word to remove: ")

new\_string = remove\_character(string, char)

print("String after removal:", new\_string)

### Task 29: Remove the Characters Which Have Odd Index Values of a Given String

def remove\_odd\_index\_chars(string):

return ''.join([char for index, char in enumerate(string) if index % 2 == 0])

string = input("Enter a string: ")

new\_string = remove\_odd\_index\_chars(string)

print("String after removing characters with odd index values:", new\_string)

### Task 30: Reverse a String If Its Length Is a Multiple of 5

def reverse\_if\_multiple\_of\_5(string):

if len(string) % 5 == 0:

return string[::-1]

return string

string = input("Enter a string: ")

new\_string = reverse\_if\_multiple\_of\_5(string)

print("Transformed string:", new\_string)

### Task 31: Format a Number with a Percentage

number = float(input("Enter a number: "))

percentage = "{:.0%}".format(number)

print("Formatted percentage:", percentage)

### Task 32: Reverse Words in a String

string = input("Enter a string: ")

words = string.split()

reversed\_words = ' '.join(reversed(words))

print("Reversed words in string:", reversed\_words)

### Task 33: Swap Cases of a Given String

string = input("Enter a string: ")

swapped\_string = string.swapcase()

print("String with swapped cases:", swapped\_string)

### Task 34: Remove Spaces from a Given String

string = input("Enter a string: ")

no\_spaces = string.replace(" ", "")

print("String without spaces:", no\_spaces)

### Task 35: Remove Duplicate Characters of a Given String

def remove\_duplicates(string):

seen = set()

new\_string = ''.join([char for char in string if not (char in seen or seen.add(char))])

return new\_string

string = input("Enter a string: ")

unique\_string = remove\_duplicates(string)

print("String with duplicate characters removed:", unique\_string)

### Task 36: Find the Area of a Circle

import math

radius = float(input("Enter the radius of the circle: "))

area = math.pi \* radius \* radius

print("Area of the circle:", area)

### Task 37: Find Sum of Squares of First n Natural Numbers

n = int(input("Enter a number: "))

sum\_of\_squares = sum(i\*\*2 for i in range(1, n+1))

print("Sum of squares of first", n, "natural numbers:", sum\_of\_squares)

### Task 38: Find Cube Sum of First n Natural Numbers

n = int(input("Enter a number: "))

cube\_sum = sum(i\*\*3 for i in range(1, n+1))

print("Cube sum of first", n, "natural numbers:", cube\_sum)

### Task 39: Find Simple Interest and Compound Interest

def simple\_interest(principal, rate, time):

return (principal \* rate \* time) / 100

def compound\_interest(principal, rate, time):

return principal \* (pow((1 + rate / 100), time))

principal = float(input("Enter the principal amount: "))

rate = float(input("Enter the rate of interest: "))

time = float(input("Enter the time period in years: "))

si = simple\_interest(principal, rate, time)

ci = compound\_interest(principal, rate, time) - principal

print("Simple Interest:", si)

print("Compound Interest:", ci)

### Task 40: Check Whether a Number Is Prime or Not

def is\_prime(num):

if num <= 1:

return False

for i in range(2, int(num\*\*0.5) + 1):

if num % i == 0:

return False

return True

num = int(input("Enter a number: "))

if is\_prime(num):

print(num, "is a prime number.")

else:

print(num, "is not a prime number.")