07-functions-and-methods-execrise

July 23, 2024

1 Functions and Methods Homework

Complete the following questions: _____ **Write a function that computes the volume of a sphere given its radius.**

volume = 4/3 pi r^3

[1]:
```
import math

def vol(rad):
    # Compute the volume of the sphere
    volume = (4/3) * math.pi * (rad ** 3)
    return volume

    # code here
```

Write a function that checks whether a number is in a given range (Inclusive of high and low)

[2]:
```
def ran_check(num, low, high):
    # Check if the number is within the range inclusive of low and high
    if low <= num <= high:
        return True
    else:
        return False
```

If you only wanted to return a boolean:

[3]:
```
def ran_bool(num, low, high):
    # Return True if the number is within the range inclusive of low and high,␣
 ↪otherwise False
    return low <= num <= high
```

[4]:
```
ran_bool(3,1,10)
```

[4]:
```
True
```

Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters.

```
Sample String : 'Hello Mr. Rogers, how are you this fine Tuesday?'
Expected Output :
No. of Upper case characters : 4
No. of Lower case Characters : 33
```

If you feel ambitious, explore the Collections module to solve this problem!

[5]:
```python
def up_low(s):
    # Initialize counters
    upper_count = 0
    lower_count = 0

    # Iterate through each character in the string
    for char in s:
        if char.isupper():
            upper_count += 1
        elif char.islower():
            lower_count += 1

    # Print the results
    print(f"No. of Upper case characters : {upper_count}")
    print(f"No. of Lower case characters : {lower_count}")

# Example usage
sample_string = 'Hello Mr. Rogers, how are you this fine Tuesday?'
up_low(sample_string)
```

```
No. of Upper case characters : 4
No. of Lower case characters : 33
```

Write a Python function that takes a list and returns a new list with unique elements of the first list.

```
Sample List : [1,1,1,1,2,2,3,3,3,3,4,5]
Unique List : [1, 2, 3, 4, 5]
```

[6]:
```python
def unique_list(l):
    # Use set to get unique elements and convert it back to a list
    unique_elements = list(set(l))
    return unique_elements
```

[7]:
```python
unique_list([1,1,1,1,2,2,3,3,3,3,4,5])
```

[7]:
```
[1, 2, 3, 4, 5]
```

Write a Python function to multiply all the numbers in a list.

```
Sample List : [1, 2, 3, -4]
Expected Output : -24
```

[8]:
```python
def multiply(numbers):
    # Initialize the result to 1
    result = 1

    # Iterate through each number in the list
    for num in numbers:
        result *= num  # Multiply each number with the result

    return result
```

[9]:
```python
multiply([1,2,3,-4])
```

[9]:
```
-24
```

Write a Python function that checks whether a passed string is palindrome or not.

Note: A palindrome is word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run.

[10]:
```python
def palindrome(s):
    # Remove spaces and convert to lowercase for a consistent check
    cleaned_s = s.replace(" ", "").lower()

    # Check if the cleaned string is equal to its reverse
    return cleaned_s == cleaned_s[::-1]
```

[11]:
```python
palindrome('helleh')
```

[11]:
```
True
```

1.1 Given a number N.Find Sum of 1 to N Using Recursion

[12]:
```python
def summation(n):
    # Base case: if n is 1, return 1
    if n == 1:
        return 1
    # Recursive case: n + sum of numbers from 1 to n-1
    else:
        return n + summation(n - 1)

# Example usage
n = int(input)
```

```
sum = summation(n)
print(sum)
```

```
Enter a number: 10
55
```

1.1.1 Define a function which can generate and print a list where the values are square of numbers between 1 and 20

[13]:
```
def printList():
    # Generate a list with the square of numbers from 1 to 20
    squares = [i**2 for i in range(1, 21)]

    # Print the generated list
    print(squares)

# Call the function to print the list
printList()
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324,
361, 400]
```

1.1.2 Define a function which can generate a dictionary where the keys are numbers between 1 and 20 (both included) and the values are square of keys. The function should just print the keys only.

[14]:
```
def printDict():
    # Generate the dictionary with keys and their squares as values
    square_dict = {i: i**2 for i in range(1, 21)}

    # Print only the keys of the dictionary
    print(square_dict.keys())

# Call the function to print the keys
printDict()
```

```
dict_keys([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20])
```

1.1.3 Write a function that count the no of characters of the given input text

- input
  - innomatics research labs
- output
  - innomatics :10
  - research : 8
  - labs : 4

[15]:
```
def count_characters(text):
    # Split the text into words
    words = text.split()

    # Iterate over each word and print its length
    for word in words:
        print(f"{word} : {len(word)}")

# Example usage
input_text = 'innomatics research labs'
count_characters(input_text)
```

```
innomatics : 10
research : 8
labs : 4
```

1.1.4 Write a program which can map() to make a list whose elements are square of elements in [1,2,3,4,5,6,7,8,9,10].

Using map() function

[16]:
```
# List of numbers
li = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# Define a function that squares a number
def square(x):
    return x ** 2

# Use map() to apply the square function to each element in the list
squaredNumbers = map(square, li)

# Convert the map object to a list and print it
print(list(squaredNumbers))
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

1.1.5 Write a program which can map() and filter() to make a list whose elements are square of even number in [1,2,3,4,5,6,7,8,9,10].

using filter()

[17]:
```
# Function to check if a number is even
def even(x):
    return x % 2 == 0

# Function to square a number
def square(x):
    return x ** 2
```

```
# List of numbers
li = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# Filter out even numbers and then square them
squared_even_numbers = map(square, filter(even, li))

# Convert the result to a list and print it
print(list(squared_even_numbers))
```

```
[4, 16, 36, 64, 100]
```

1.1.6 Write a program which can filter() to make a list whose elements are even number between 1 and 20 (both included)

[18]:
```
# Function to check if a number is even
def even(x):
    return x % 2 == 0

# Use filter() to get even numbers in the range from 1 to 20
evenNumbers = filter(even, range(1, 21))

# Convert the filter object to a list and print it
print(list(evenNumbers))
```

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

1.1.7 $f(n) = f(n-1) + f(n-2)$ if n>1

[19]:
```
def f(n):
    # Base cases
    if n == 0:
        return 0
    elif n == 1:
        return 1
    # Recursive case
    else:
        return f(n - 1) + f(n - 2)

# Get user input
n = int(input("Enter n value: "))

# Print the nth Fibonacci number
print(f(n))
```

```
Enter n value: 7
13
```

[ ]:

2 Innomatics Research Labs

www.innomatics.in

[ ]: