

# Python Functions

---

1. Write a function to calculate the factorial of a number (non-recursive).

```
2.1.py > ...
1  def factorial(n):
2      result = 1
3      for i in range(2, n + 1):
4          result = result*i
5      return result
6
7  number = int(input("Enter a number to find factorial: "))
8  print(f"Factorial of {number} is {factorial(number)}")
```

2. Define a function that checks whether a string is a palindrome.

```
2.2.py > ...
1  def is_palindrome(s):
2      s = s.lower().replace(" ", "")
3      left = 0
4      right = len(s) - 1
5
6      while left < right:
7          if s[left] != s[right]:
8              return False
9          left += 1
10         right -= 1
11     return True
12
13 string = input("Enter a string: ")
14 print(f"The string '{string}', is a Palindrome!" if is_palindrome(string)
15       else f"The string '{string}', is not a palindrome.")
```

3. Write a function that accepts a list and returns the sum and average of the numbers.

```
2.3.py > ...
1  def sumAndAverage(list):
2      total = sum(list)
3      average = total / len(list) if list else 0
4      print(f"Sum = {total}, Average = {average}")
5
6  list = list(map(int, input("Enter numbers separated by space: ").split()))
7  sumAndAverage(list)
```

4. Create a function that returns the nth Fibonacci number using recursion.

```
2.4.py > ...
1  def fibonacci(n):
2      if n <= 1:
3          return n
4      return fibonacci(n - 1) + fibonacci(n - 2)
5
6  term = int(input("Enter which Fibonacci term to print: "))
7  print(f"{term}th Fibonacci number is {fibonacci(term)}")
```

5. Define a function to count the number of vowels in a given string.

```
2.5.py > ...
1  def count_vowels(s):
2      vowels = "aeiouAEIOU"
3      sum = 0
4      for x in s :
5          if x in vowels : sum += 1
6      return sum
7
8  string = input("Enter a string: ")
9  print(f"Number of vowels in {string}: {count_vowels(string)}")
```

6. Implement a decorator that measures execution time of any function.

7. Write a recursive function to solve the Tower of Hanoi problem.

8. Implement a function that uses variable-length arguments to sum any number of inputs.

9. Write a function that flattens a nested list using recursion.

10. Implement a memoized version of the Fibonacci sequence.