Exercise 1: Write a function that prints 'Hello, World!'

```
def hello():
   print("Hello, World!")
hello()
```

Exercise 2: Write a function that takes a name as input and prints a greeting

```
def greet(name):
   print("Hello", name)
greet("Lokesh")
```

Exercise 3: Write a function that returns the square of a number

```
def square(n):
   return n * n
print(square(5))
```

Exercise 4: Write a function to add two numbers and return the result

```
def add(a, b):
   return a + b
print(add(3, 4))
```

Exercise 5: Write a function that returns the maximum of three numbers

```
def maximum(a, b, c):
   return max(a, b, c)
print(maximum(3, 7, 5))
```

Exercise 6: Write a function that checks if a number is even or odd

```
def even_odd(n):
   return "Even" if n % 2 == 0 else "Odd"
print(even_odd(6))
```

Exercise 7: Write a function that takes a number and returns True if it is prime

```
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True
print(is_prime(11))
```

Exercise 8: Write a function to calculate the factorial of a number

```
def factorial(n):
    result = 1
    for i in range(2, n + 1):
        result *= i
    return result
print(factorial(5))
```

Exercise 9: Write a function to find the sum of all numbers in a list

```
def sum_list(lst):
    return sum(lst)
print(sum_list([1, 2, 3, 4, 5]))
```

Exercise 10: Write a function that accepts a list and returns the largest number

```
def max_list(lst):
    return max(lst)
print(max_list([10, 45, 2, 89]))
```

Exercise 11: Write a function that greets a user with a default name if no name is provided

```
def greet_default(name="User"):
    print("Hello", name)
greet_default()
greet_default("Lokesh")
```

Exercise 12: Write a function with two arguments and use keyword arguments to call it

```
def describe(name, age):
    print(name, "is", age, "years old")
describe(age=22, name="Lokesh")
```

Exercise 13: Write a recursive function to calculate factorial

```
def fact(n):
    if n == 0 or n == 1:
        return 1
    return n * fact(n - 1)
print(fact(5))
```

Exercise 14: Write a recursive function to print Fibonacci sequence up to n terms

```
def fib(n):
    if n <= 1:
        return n
    return fib(n - 1) + fib(n - 2)
for i in range(7):
    print(fib(i), end=" ")
```

Exercise 15: Use a lambda function to double a number

```
double = lambda x: x * 2
print(double(5))
```

Exercise 16: Use map() to square all numbers in a list

```
nums = [1, 2, 3, 4]
squares = list(map(lambda x: x * x, nums))
print(squares)
```

Exercise 17: Use filter() to get even numbers from a list

```
evens = list(filter(lambda x: x % 2 == 0, nums))
print(evens)
```

Exercise 18: Write a function that returns True if a string is a palindrome

```
def is_palindrome(s):
   return s == s[::-1]
print(is_palindrome("madam"))
```

Exercise 19: Write a function to count vowels in a string

```
def count_vowels(s):
   return sum(1 for ch in s if ch.lower() in 'aeiou')
print(count_vowels("hello world"))
```

Exercise 20: Write a function that takes a string and returns a dictionary with character counts

```
def char_count(s):
   d = {}
   for ch in s:
      d[ch] = d.get(ch, 0) + 1
   return d
print(char_count("banana"))
```