20 Python Functions

This document provides 20 Python functions, including their solutions and expected results.

Easy Functions

1. Create a Function

Write a function that returns “Hello, World!”.

```python
def hello_world():
    return "Hello, World!"

# Test
print(hello_world())  # Output: "Hello, World!"
```

2. Add Two Numbers

Write a function to add two numbers.

```python
def add(a, b):
    return a + b

# Test
print(add(2, 3))  # Output: 5
```

3. Subtract Two Numbers

Write a function to subtract two numbers.

```python
def subtract(a, b):
    return a - b

# Test
print(subtract(5, 3))  # Output: 2
```

4. Multiply Two Numbers

Write a function to multiply two numbers.

```python
def multiply(a, b):
    return a * b

# Test
print(multiply(2, 3))  # Output: 6
```

5. Divide Two Numbers

Write a function to divide two numbers.

```python
def divide(a, b):
    return a / b

# Test
print(divide(6, 3))  # Output: 2.0
```

Medium Functions

6. Calculate Factorial

Write a function to calculate the factorial of a number.

```python
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

# Test
print(factorial(5))  # Output: 120
```

7. Fibonacci Sequence

Write a function to generate the Fibonacci sequence up to n terms.

```python
def fibonacci(n):
    sequence = [0, 1]
    while len(sequence) < n:
        sequence.append(sequence[-1] + sequence[-2])
    return sequence

# Test
print(fibonacci(5))  # Output: [0, 1, 1, 2, 3]
```

8. Check Prime

Write a function to check if a number is prime.

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

# Test
print(is_prime(7))  # Output: True
```

9. Find GCD

Write a function to find the greatest common divisor (GCD) of two numbers.

```
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a

# Test
print(gcd(48, 18))  # Output: 6
```

10. Find LCM

Write a function to find the least common multiple (LCM) of two numbers.

```
def lcm(a, b):
    return abs(a*b) // gcd(a, b)

# Test
print(lcm(4, 5))  # Output: 20
```

11. Reverse String

Write a function to reverse a string.

```
def reverse_string(s):
    return s[::-1]

# Test
print(reverse_string("hello"))  # Output: "olleh"
```

12. Count Vowels

Write a function to count the number of vowels in a string.

```
def count_vowels(s):
    return sum(1 for char in s if char.lower() in 'aeiou')

# Test
print(count_vowels("hello"))  # Output: 2
```

13. Check Palindrome

Write a function to check if a string is a palindrome.

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

```python
# Test
print(is_palindrome("racecar"))  # Output: True
```

14. Convert to Uppercase

Write a function to convert a string to uppercase.

```python
def to_uppercase(s):
    return s.upper()

# Test
print(to_uppercase("hello"))  # Output: "HELLO"
```

15. Remove Duplicates

Write a function to remove duplicates from a list.

```python
def remove_duplicates(lst):
    return list(set(lst))

# Test
print(remove_duplicates([1, 2, 2, 3, 3, 3, 4]))  # Output: [1, 2, 3, 4]
```

16. Find Length

Write a function to find the length of a list.

```python
def find_length(lst):
    return len(lst)

# Test
print(find_length([1, 2, 3, 4, 5]))  # Output: 5
```

17. Find Max

Write a function to find the maximum element in a list.

```python
def find_max(lst):
    return max(lst)

# Test
print(find_max([1, 2, 3, 4, 5]))  # Output: 5
```

18. Find Min

Write a function to find the minimum element in a list.

```python
def find_min(lst):
    return min(lst)
```

```python
# Test
print(find_min([1, 2, 3, 4, 5]))  # Output: 1
```

19. List Comprehension

Write a function that returns a list of squares of the elements in a list.

```python
def list_comprehension(lst):
    return [x ** 2 for x in lst]

# Test
print(list_comprehension([1, 2, 3]))  # Output: [1, 4, 9]
```

20. Find Common Elements

Write a function to find common elements in multiple lists.

```python
def common_elements(*lists):
    return set.intersection(*map(set, lists))

# Test
print(common_elements([1, 2, 3], [2, 3, 4], [3, 4, 5]))  # Output: {3}
```

Stay Updated

Be sure to this repository to stay updated with new examples and enhancements!

License

This project is protected under the MIT License.

Contact

Panagiotis Moschos - pan.moschos86@gmail.com

Note: This is a Python script and requires a Python interpreter to run.

Happy Coding

Made with by Panagiotis Moschos (https://github.com/pmoschos)