Python Coding Interview Cheat Sheet

BY - ADITYA KALYAN | DATA ANALYST

Reverse a String

```
s = "hello"
print(s[::-1])  # Output: "olleh"
```

Check if a String is a Palindrome

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

Find Duplicates in a List

```
def find_duplicates(lst):
    return list(set([x for x in lst if lst.count(x) > 1]))
print(find_duplicates([1, 2, 3, 2, 4, 5, 5]))
```

Find the Missing Number in a List

```
def missing_number(nums):
    n = len(nums) + 1
    return n * (n - 1) // 2 - sum(nums)
print(missing_number([0, 1, 3]))
```

Find the First Non-Repeating Character

```
from collections import Counter

def first_unique(s):
    count = Counter(s)
    for char in s:
        if count[char] == 1:
            return char
    return None
print(first_unique("swiss"))
```

Check if Two Strings are Anagrams

```
from collections import Counter

def is_anagram(s1, s2):
    return Counter(s1) == Counter(s2)
print(is_anagram("listen", "silent"))
```

Find the Intersection of Two Lists

```
def list_intersection(lst1, lst2):
    return list(set(lst1) & set(lst2))
print(list_intersection([1, 2, 3], [2, 3, 4]))
```

Fibonacci Sequence (Recursion & Iterative)

```
def fibonacci(n):
    if n <= 1:
        return n
    return fibonacci(n - 1) + fibonacci(n - 2)
print(fibonacci(6))
```

Find Factorial (Recursion & Iterative)

```
def factorial(n):
    return 1 if n == 0 else n * factorial(n - 1)
print(factorial(5))
```

Find the Most Frequent Element in a List

```
from collections import Counter

def most_frequent(lst):
    return Counter(lst).most_common(1)[0][0]
print(most_frequent([1, 3, 2, 1, 4, 1]))
```

Merge Two Sorted Lists

```
def merge_sorted_lists(lst1, lst2):
    return sorted(lst1 + lst2)
print(merge_sorted_lists([1, 3, 5], [2, 4, 6]))
```

Remove Duplicates from a List

```
def remove_duplicates(lst):
    return list(set(lst))
print(remove_duplicates([1, 2, 2, 3, 4, 4, 5]))
```

Find Pairs That Sum to a Target Value

```
def find_pairs(nums, target):
    seen = set()
    pairs = []
    for num in nums:
        if target - num in seen:
            pairs.append((num, target - num))
        seen.add(num)
    return pairs
```

```
print(find_pairs([1, 2, 3, 4, 5], 6))
```

Check if a Number 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(7))
```

Flatten a Nested List

```
def flatten_list(nested_list):
    result = []
    for i in nested_list:
        if isinstance(i, list):
            result.extend(flatten_list(i))
        else:
            result.append(i)
    return result
print(flatten_list([[1, 2], [3, [4, 5]]]))
```

Find Second Largest Element in a List

```
def second_largest(lst):
    unique_sorted = sorted(set(lst), reverse=True)
    return unique_sorted[1] if len(unique_sorted) > 1 else None
print(second_largest([1, 2, 3, 4, 4]))
```

Find Common Elements in Three Sorted Lists

```
def common_elements(lst1, lst2, lst3):
    return list(set(lst1) & set(lst2) & set(lst3))
print(common_elements([1, 2, 3], [2, 3, 4], [3, 4, 5]))
```

Generate All Permutations of a String

```
from itertools import permutations

def string_permutations(s):
    return ["".join(p) for p in permutations(s)]
print(string_permutations("abc"))
```

Find the Longest Word in a Sentence

```
def longest_word(sentence):
    words = sentence.split()
    return max(words, key=len)
print(longest_word("Python is an amazing programming language"))
```

Count the Occurrences of Each Character in a String

```
from collections import Counter

def char_count(s):
    return dict(Counter(s))
print(char_count("hello"))
```