* PYTHON CHEATSHEET 🚨 FOR JOB INTERVIEWS 📝*

1. Functions: Define and Use Functions

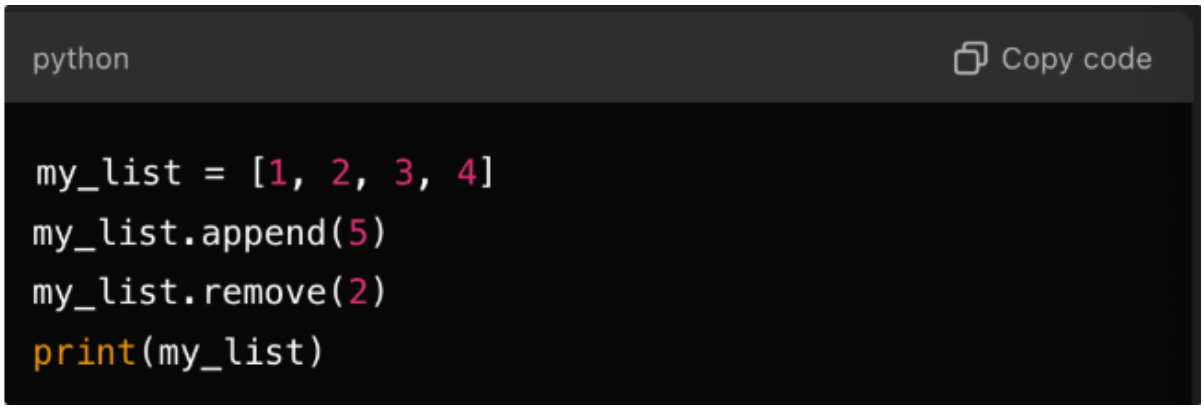
Explanation: Defines a reusable function to greet a user and demonstrates its use.

Code Example:

2. Lists: Basic Operations on Lists

Explanation: Shows how to add, remove, and manipulate elements in a list.

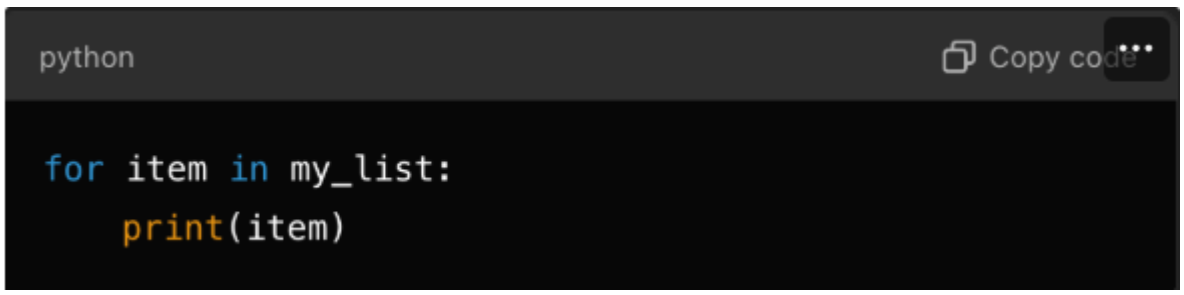
Code Example:



3. Loops: Iterating Over Lists

Explanation: Iterates over a list and prints each item.

Code Example:



4. File Handling: Read and Write Files

Explanation: Demonstrates writing to and reading from a text file using the 'with' statement.

Code Example:

```python
with open('example.txt', 'w') as file:
    file.write('Hello, world!')

with open('example.txt', 'r') as file:
    content = file.read()
    print(content)
```

5. Data Manipulation with Dictionaries

Explanation: Adds and accesses elements in a dictionary, demonstrating data manipulation.

Code Example:

```python
my_dict = {'name': 'Anurag', 'age': 30}
my_dict['location'] = 'Bangalore'
print(my_dict['name'])
```

6. List Comprehensions

Explanation: Creates a list of squares using list comprehension.

Code Example:

```python
squares = [x**2 for x in range(5)]
print(squares)
```

7. Exception Handling

Explanation: Handles division by zero error using a try-except block.

Code Example:

```python
try:
    result = 10 / 0
except ZeroDivisionError as e:
    print(f'Error: {e}')
```

8. Reading CSV Files

Explanation: Reads and prints rows from a CSV file using the csv module.

Code Example:

```python
import csv

with open('data.csv', 'r') as csvfile:
    reader = csv.reader(csvfile)
    for row in reader:
        print(row)
```

9. Writing JSON Data

Explanation: Writes a dictionary to a JSON file using the json module.

Code Example:

```python
import json

data = {'name': 'Anurag', 'age': 30}
with open('data.json', 'w') as jsonfile:
    json.dump(data, jsonfile)
```

10. Pandas: DataFrame Basics

Explanation: Creates a simple Pandas DataFrame and prints it.

Code Example:

```python
import pandas as pd

data = {'Name': ['Anurag', 'John'], 'Age': [30, 25]}
df = pd.DataFrame(data)
print(df)
```

PRACTICAL QUESTIONS FOR INTERVIEWS

1. Reverse a String Without Using Built-in Functions

Question:

How would you reverse a string in Python without using built-in slicing or reverse functions?

Answer:

```python
def reverse_string(s):
    reversed_s = ''
    for char in s:
        reversed_s = char + reversed_s
    return reversed_s

print(reverse_string("Python"))
```

2. Check if Two Strings are Anagrams

Question:

Write a function to check if two strings are anagrams of each other.

Answer:

```python
def are_anagrams(str1, str2):
    return sorted(str1) == sorted(str2)

print(are_anagrams("listen", "silent"))  # True
```

3. Find the Missing Number in a List

Question:

Given a list of integers from 1 to n with one number missing, find the missing number.

Answer:

```python
def find_missing_number(nums, n):
    expected_sum = n * (n + 1) // 2
    actual_sum = sum(nums)
    return expected_sum - actual_sum

nums = [1, 2, 3, 5]
print(find_missing_number(nums, 5))  # 4
```

4. Flatten a Nested List

Question:

Write a function to flatten a nested list.

Answer:

```python
def flatten_list(nested_list):
    flat_list = []
    for item in nested_list:
        if isinstance(item, list):
            flat_list.extend(flatten_list(item))
        else:
            flat_list.append(item)
    return flat_list

nested_list = [1, [2, [3, 4]], 5]
print(flatten_list(nested_list))  # [1, 2, 3, 4, 5]
```

5. Implement a Simple Cache (Memoization)

Question:

Write a function to cache the results of expensive computations.

Answer:

```python
def fibonacci(n, cache={}):
    if n in cache:
        return cache[n]
    if n <= 2:
        return 1
    cache[n] = fibonacci(n - 1, cache) + fibonacci(n - 2, cache)
    return cache[n]

print(fibonacci(10))  # 55
```

6. Find the First Non-Repeating Character

Question:

Write a function to find the first non-repeating character in a string.

Answer:

```python
def first_non_repeating_char(s):
    char_count = {}
    for char in s:
        char_count[char] = char_count.get(char, 0) + 1
    for char in s:
        if char_count[char] == 1:
            return char
    return None

print(first_non_repeating_char("swiss"))  # w
```

7. Sort a Dictionary by Its Values

Question:

How would you sort a dictionary by its values?

Answer:

```python
def sort_dict_by_values(d):
    return dict(sorted(d.items(), key=lambda item: item[1]))

d = {'apple': 3, 'banana': 1, 'cherry': 2}
print(sort_dict_by_values(d))  # {'banana': 1, 'cherry': 2, 'apple': 3
```

8. Find the Longest Palindromic Substring

Question:

Write a function to find the longest palindromic substring in a given string.

Answer:

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

    max_palindrome = ''
    for i in range(len(s)):
        for j in range(i + 1, len(s) + 1):
            sub = s[i:j]
            if is_palindrome(sub) and len(sub) > len(max_palindrome):
                max_palindrome = sub
    return max_palindrome

print(longest_palindrome("babad"))  # "bab" or "aba"
```

9. Count the Frequency of Words in a String

Question:

Write a function to count the frequency of each word in a string.

Answer:

```python
def word_frequency(s):
    words = s.split()
    frequency = {}
    for word in words:
        frequency[word] = frequency.get(word, 0) + 1
    return frequency

print(word_frequency("the quick brown fox jumps over the lazy dog the fox"))
# {'the': 3, 'quick': 1, 'brown': 1, 'fox': 2, 'jumps': 1, 'over': 1, 'lazy': 1, '
```

10. Generate All Permutations of a List

Question:

Write a function to generate all permutations of a list.

Answer:

python Copy code

```python
from itertools import permutations

def generate_permutations(lst):
    return list(permutations(lst))

print(generate_permutations([1, 2, 3]))
# [(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)]
```

----------END----------