

# Python Questions

## 1. Generate a list of even numbers from 1 to 10?

```
even_numbers = [i for i in range(1, 11) if i % 2 == 0]
print(even_numbers)
```

## 2. Remove duplicates from list = [1,2,3,4,5,6,7,8,9,10]

(No actual duplicates here, but for example:)

```
nums = [1,2,3,4,5,6,7,8,9,10]
unique_nums = list(set(nums))
print(unique_nums)
```

## 3. Reverse a string name = "VAARAHI"?

```
name = "VAARAHI"
reversed_name = name[::-1]
print(reversed_name)
```

## 4. Find word count in: "I love my country"?

```
sentence = "I love my country"
word_count = len(sentence.split())
print(word_count)
```

## 5. Write a break statement for 10 numbers at position 7?

```
for i in range(1, 11):
    if i == 7:
        break
    print(i)
```

## 6. Write a continue statement for 10 numbers at position 7?

```
for i in range(1, 11):
    if i == 7:
        continue
    print(i)
```

## 7. Write a Python function to 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
```

# Python Questions

## 8. Check 1-10 numbers if Odd or Even in a function

```
def check_odd_even():
    for i in range(1, 11):
        if i % 2 == 0:
            print(f"{i} is Even")
        else:
            print(f"{i} is Odd")
check_odd_even()
```

## 9. Print all numbers from 1 to 10 using range

```
for i in range(1, 11):
    print(i)
```

## 10. Print the first 5 multiples of 3

```
for i in range(1, 6):
    print(i * 3)
```

## 11. Concatenate strings

```
String1 = "Hi"
String2 = " How are you"
result = String1 + String2
print(result)
```

## 12. Update the age and print the updated dictionary

```
person = {'name': 'John', 'age': 25, 'city': 'New York'}
person['age'] = 30
print(person)
```

## 13. Use f-strings to print: The product of 5 and 7 is 35

```
a = 5
b = 7
print(f"The product of {a} and {b} is {a * b}")
```

## 14. Reverse order of words in a sentence

```
sentence = "Python is amazing"
reversed_sentence = ' '.join(sentence.split()[::-1])
print(reversed_sentence)
```

## 15. Use lambda to create squares from 1 to 5

```
squares = list(map(lambda x: x**2, range(1, 6)))
print(squares)
```

# Python Questions

## 16. Function to return unique elements from a list

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

## 17. Function to find largest element without built-in functions

```
def find_largest(lst):  
    largest = lst[0]  
    for num in lst:  
        if num > largest:  
            largest = num  
    return largest  
  
print(find_largest([10, 20, 5, 30]))
```

## 18. Remove duplicates from a list while maintaining order

```
def remove_duplicates(lst):  
    result = []  
    seen = set()  
    for item in lst:  
        if item not in seen:  
            seen.add(item)  
            result.append(item)  
    return result  
  
print(remove_duplicates([1, 2, 2, 3, 1, 4]))
```

## 19. Find the missing number in sequence from 1 to n

```
def find_missing_number(numbers, n):  
    expected_sum = n * (n + 1) // 2  
    actual_sum = sum(numbers)  
    return expected_sum - actual_sum  
  
print(find_missing_number([1, 2, 4, 5], 5)) # Output: 3
```

## 20. Print pattern using nested loops

```
*  
**  
***  
****  
*****  
  
for i in range(1, 6):  
    for j in range(i):  
        print("*", end="")  
    print()
```

# Python Questions

## 21. Print diamond pattern using nested loops

```
  *
 ***
*****
*****
*****
*****
 ***
  *
```

```
n = 5
# Upper part
for i in range(n):
    print(" " * (n - i - 1) + "*" * (2*i + 1))
# Lower part
for i in range(n-2, -1, -1):
    print(" " * (n - i - 1) + "*" * (2*i + 1))
```

## 22. Function to return True if number is prime

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

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

## 23. Capitalize the first letter of a string

```
s = "python is great"
print(s.capitalize())
```

## 24. Function that accepts keyword arguments and prints them

```
python
CopyEdit
def print_kwargs(**kwargs):
    for key, value in kwargs.items():
        print(f"{key} = {value}")
```

```
print_kwargs(name="Harish", age=30)
```

## 25. Use a global variable within a function

```
count = 0
```

```
def increment():
    global count
    count += 1
```

```
increment()
print(count)
```

# Python Questions

## 26. Find the length of a list

```
lst = [10, 20, 30, 40, 50]
print(len(lst))
```

## 27. Swap first and last elements of a list

```
lst = [1, 2, 3]
lst[0], lst[-1] = lst[-1], lst[0]
print(lst)
```

## 28. How to clear a list in different methods?

```
lst = [1, 2, 3]

# Method 1: Using clear()
lst.clear()

# Method 2: Reinitialization
lst = []

# Method 3: Using del
del lst[:]
```

## 29. Find second largest number from list

```
mylist = [90, 11, 20, 40, 100]
mylist.sort()
print(mylist[-2]) # Output: 90
```

## 30. Arithmetic operations

```
a = 10
b = 3
print("Addition:", a + b)
print("Subtraction:", a - b)
print("Multiplication:", a * b)
print("Division:", a / b)
print("Floor Division:", a // b)
print("Modulus:", a % b)
print("Exponent:", a ** b)
```

## 31. Find middle word using index

```
text = "This is my interview python question"
words = text.split()
middle = words[len(words) // 2]
print(middle) # Output: interview
```

# Python Questions

## 32. Typecasting data types

```
print(float(10))          # int to float
print(int(10.6))          # float to int
print(complex(10))        # int to complex
print(complex(10.5))      # float to complex
print(float(complex(10)))  # complex to float not directly possible,
                           # gives error
```

## 33. Find below output

```
input_list = ["HI", "hello", (1,2,3,4), "a", "b"]
count = 0
for item in input_list:
    if isinstance(item, str) and len(item) == 1:
        count += 1
print(count)  # Output: 3
```

## 34. Extract details from code

```
details = "INDTGHYD500082"
print("COUNTRY:", details[0:3])
print("STATE:", details[3:5])
print("CITY:", details[5:8])
print("PINCODE:", details[8:])
```

## 35. Convert item dictionary to required format

```
Item = {'mobile': 300, 'laptop': 400, 'tab': 200}
output = {
    'ITEM': 'mobile',
    'QUANTITY': 3,
    'PRICE': Item['mobile'],
    'TOTAL_PRICE': Item['mobile'] * 3
}
print(output)
```

## 36. Print city name using if-else

```
cities = ['hyd', 'Chennai', 'Bangalore']
for city in cities:
    if city == 'hyd':
        print("Hyderabad")
    elif city == 'Chennai':
        print("Chennai")
    elif city == 'Bangalore':
        print("Bangalore")
```

# Python Questions

## 37. Convert string case

```
s = "Welcome to Vaarahi"
print(s.upper())      # ALL UPPERCASE
print(s.lower())      # all lowercase
print(s.swapcase())   # SWAPCASE
```

## 38. Count how many times 2 appears

```
lst = [1, 2, 3, 4, 2, 5, 2, 6, 2, 6, 2]
print(lst.count(2))   # Output: 5
```

## 39. Clear list and remove last element

```
lst = [1, 2, 3, 4]
lst.clear()
# or if you want to remove just the last element:
lst = [1, 2, 3, 4]
lst.pop()
print(lst)   # Output: [1, 2, 3]
```

## 40. Remove 'apple' from list

```
python
CopyEdit
fruits = ['apple', 'mango', 'orange']
fruits.remove('apple')
print(fruits)   # Output: ['mango', 'orange']
```

## 41. Sort dictionary by values

```
input_dict = {'ravi': 10, 'rajnish': 9, 'sanjeev': 15, 'yash': 2, 'suraj': 32}
sorted_dict = dict(sorted(input_dict.items(), key=lambda item: item[1]))
print(sorted_dict)
```

*ChatGPT said:*

Here are the Python answers for questions **42 to 50** based on the fourth image:

## 42. Sum of all dictionary values

```
python
CopyEdit
d = {'a': 100, 'b': 200, 'c': 300}
print(sum(d.values()))   # Output: 600
```

# Python Questions

## 43. Get dictionary keys as a list

```
d1 = {1: 'a', 2: 'b', 3: 'c'}
d2 = {'A': 'ant', 'B': 'ball'}
print(list(d1.keys())) # Output: [1, 2, 3]
print(list(d2.keys())) # Output: ['A', 'B']
```

## 44. Swap first and last elements of a list

```
lst = [12, 35, 9, 56, 24]
lst[0], lst[-1] = lst[-1], lst[0]
print(lst) # Output: [24, 35, 9, 56, 12]
```

## 45. Check if 3 exists in list

```
test_list = [1, 6, 3, 5, 3, 4]
print(3 in test_list) # Output: True
```

## 46. Find common elements in two lists

```
a = [2, 3, 5, 6, 7]
b = [1, 9, 8] # Assuming second list to compare with
print(list(set(a) & set(b))) # Output: [] (no common elements)
```

## 47. Merge two dictionaries

```
dict1 = {'a': 10, 'b': 8}
dict2 = {'d': 6, 'c': 4}
merged = {**dict1, **dict2}
print(merged)
# Output: {'a': 10, 'b': 8, 'd': 6, 'c': 4}
```

## 48–50. Concatenate list + string, convert to tuple, return the tuple

```
lst = ["india", "is"]
string = "best"

# Step 1: Concatenate using +
concat = lst + [string]

# Step 2: Convert to tuple
result = tuple(concat)

# Step 3: Return tuple
print(result) # Output: ('india', 'is', 'best')
```