

**Malla Reddy College of Engineering & Technology**

(Autonomous Institution- UGC, Govt. of India)

(Affiliated to JNTUH, Hyderabad, Approved by AICTE, NBA &NAAC with ‘A’ Grade)

**Python Programming Guide**

**Python Technical Questions**

1. **What are the key features of Python?**
   * Python is interpreted, high-level, dynamically typed, object-oriented, and has extensive libraries.
2. **What are standard data types in Python?**
   * int, float, bool, complex, str, list, tuple, set, dict.
3. **How do you install Python on a system?**
   * Download from python.org, run installer, add Python to PATH.
4. **What is the difference between float and complex?**
   * Float is a real number (e.g., 3.5), complex has real and imaginary parts (e.g., 3+2j).
5. **How do you read input in Python?**
   * Using input() function, returns string.
6. **How do you output data?**
   * Using print() function.
7. **What are lists and tuples?**
   * Lists are mutable sequences; tuples are immutable.
8. **How do you convert data types in Python?**
   * Using built-in functions like int(), float(), str().
9. **What are comments in Python?**
   * Single-line: # comment, multi-line: triple quotes '''comment'''.
10. **What is the difference between a set and a dictionary?**
    * Set contains unique items, dictionary contains key-value pairs.
11. **How are variables declared in Python?**
    * No keyword needed, simply assign: x = 10.
12. **Explain different types of operators in Python.**
    * Arithmetic, Assignment, Comparison, Logical, Identity, Membership, Bitwise.
13. **What is operator precedence?**
    * Rules that determine the order of evaluation in expressions.
14. **Explain identity and membership operators.**
    * Identity: is, is not; Membership: in, not in.
15. **How do if, elif, and else work in Python?**
    * Conditional branching using indentation.
16. **How does a for loop differ from a while loop?**
    * for is for iterables; while is condition-based.
17. **What are break, continue, and pass?**
    * Control loop flow: break exits, continue skips, pass does nothing.
18. **What is indentation in Python?**
    * Whitespace to define code blocks; mandatory in Python.
19. **What is a chained conditional?**
    * Multiple conditions using if, elif, and else.
20. **How do you use range in for loops?**
    * range(start, stop, step) generates a sequence.
21. **What is NumPy and why is it used?**
    * Numerical computing library; supports large multidimensional arrays.
22. **How do you create a NumPy array?**
    * numpy.array([1, 2, 3])
23. **What is slicing in arrays?**
    * Accessing a subset: arr[1:4]
24. **How do you import the array module?**
    * import array or import numpy as np
25. **What are the types of arrays in Python?**
    * List-based arrays, array module arrays, and numpy arrays.
26. **What is array indexing?**
    * Accessing elements using their position: arr[0]
27. **What is the difference between lists and arrays?**
    * Lists are dynamic, can hold mixed types. Arrays are fixed type and more efficient for computation.
28. **What are array advantages?**
    * Faster computation, less memory, element-wise operations.
29. **How to reshape arrays in NumPy?**
    * array.reshape(new\_shape)
30. **What is the use of arange() in NumPy?**
    * Generates evenly spaced values: np.arange(0, 10, 2)
31. **How do you define a function in Python?**
    * def function\_name(parameters):
32. **What is the scope of a variable?**
    * Local inside function, global outside function.
33. **What are positional and keyword arguments?**
    * Positional are based on order; keyword use parameter names.
34. **What are default arguments?**
    * Parameters that assume a default value if not passed.
35. **What are variable-length arguments?**
    * \*args for tuple, \*\*kwargs for dictionary.
36. **What is a lambda function?**
    * Anonymous single-line function: lambda x: x\*x
37. **What is a fruitful function?**
    * Function that returns a value.
38. **What is the difference between return and print?**
    * return sends a value back; print displays it.
39. **What is a recursive function?**
    * A function that calls itself.
40. **Can lambda functions take multiple arguments?**
    * Yes: lambda x, y: x + y
41. **What are different file access modes in Python?**
    * r, w, a, r+, b, etc.
42. **How to open a file in Python?**
    * open('filename.txt', 'r')
43. **How to read from a file?**
    * file.read(), file.readline(), file.readlines()
44. **How to write to a file?**
    * file.write('text')
45. **What is the difference between text and binary files?**
    * Text is readable; binary is in byte format.
46. **How to close a file?**
    * file.close() or use with open() as file:
47. **What is exception handling in Python?**
    * Handling runtime errors using try, except, finally.
48. **What is the difference between try and finally?**
    * finally always executes, except only when exception occurs.
49. **What is the purpose of raise keyword?**
    * Manually raise an exception.
50. **What is the difference between syntax and runtime errors?**
    * Syntax errors occur during parsing; runtime errors during execution.

**Python Coding Questions**

# **1. Hello World program**

print("Hello, World!")

# **2. Input two numbers and print their sum**

a = int(input("Enter first number: "))

b = int(input("Enter second number: "))

print("Sum:", a + b)

# **3. Check even or odd**

num = int(input())

print("Even" if num % 2 == 0 else "Odd")

**# 4. Swap two numbers using temporary variable**

a, b = 10, 20

a, b = b, a

print(a, b)

**# 5. Find factorial using loop**

n = int(input())

fact = 1

for i in range(1, n + 1): fact \*= i

print(fact)

**# 6. Fibonacci series up to n terms**

n = int(input())

a, b = 0, 1

for \_ in range(n):

print(a, end=' ')

a, b = b, a + b

**# 7. Check prime number**

n = int(input())

flag = all(n % i != 0 for i in range(2, n))

print("Prime" if flag and n > 1 else "Not Prime")

**# 8. Reverse a number**

n = int(input())

rev = 0

while n:

rev = rev \* 10 + n % 10

n //= 10

print(rev)

**# 9. Count vowels in a string**

s = input()

vowels = sum(1 for ch in s if ch.lower() in 'aeiou')

print(vowels)

**# 10. Palindrome string check**

s = input()

print("Palindrome" if s == s[::-1] else "Not Palindrome")

**# 11. Sum of elements in a list**

lst = [1, 2, 3, 4]

print(sum(lst))

**# 12. Find max and min in list**

print(max(lst), min(lst))

**# 13. Count frequency of elements in list**

from collections import Counter

print(Counter(lst))

**# 14. Sort list without sort()**

print(sorted(lst))

**# 15. Merge two lists**

print([1, 2] + [3, 4])

**# 16. Tuple unpacking**

t = (1, 2, 3)

a, b, c = t

print(a, b, c)

**# 17. Set operations**

s1 = {1, 2, 3}; s2 = {3, 4, 5}

print(s1 | s2, s1 & s2)

**# 18. Dictionary operations**

d = {'a': 1, 'b': 2}

d['c'] = 3

print(d)

**# 19. List comprehension for squares**

print([x\*\*2 for x in range(5)])

**# 20. Use of zip()**

names = ['a', 'b']; scores = [10, 20]

print(dict(zip(names, scores)))

**# 21. Use of lambda and filter**

nums = [1, 2, 3, 4]

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

**# 22. Use of lambda and map**

print(list(map(lambda x: x \* 2, nums)))

**# 23. Function with default argument**

def greet(name="Guest"): print("Hello", name)

greet()

**# 24. Function with variable arguments**

def add(\*args): print(sum(args))

add(1, 2, 3)

**# 25. Global vs local**

x = 5

def f():

global x

x += 1

f(); print(x)

**# 26. Recursive factorial**

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

print(factorial(5))

**# 27. Read a file**

with open("test.txt") as f:

print(f.read())

**# 28. Write to a file**

with open("test.txt", "w") as f:

f.write("Hello World")

**# 29. Append to a file**

with open("test.txt", "a") as f:

f.write("\nMore text")

**# 30. Count words in a file**

with open("test.txt") as f:

print(len(f.read().split()))

**# 31. Exception handling**

try:

a = 10 / 0

except ZeroDivisionError:

print("Cannot divide by zero")

**# 32. Nested if**

x = int(input())

if x > 0:

if x % 2 == 0:

print("Positive Even")

**# 33. Loop with break**

for i in range(10):

if i == 5: break

print(i)

**# 34. Continue statement**

for i in range(5):

if i == 2: continue

print(i)

**# 35. Pass statement**

for i in range(3):

pass

**# 36. Use of range with step**

print(list(range(2, 10, 2)))

# 37. Bitwise operations

a, b = 5, 3

print(a & b, a | b, a ^ b)

**# 38. Identity operator**

a = [1, 2]; b = a

print(a is b)

**# 39. Membership operator**

print(1 in [1, 2, 3])

**# 40. Complex number**

a = complex(2, 3)

print(a.real, a.imag)

**# 41. Array using numpy**

import numpy as np

arr = np.array([1, 2, 3])

print(arr)

**# 42. Numpy slicing**

print(arr[1:])

**# 43. Matrix multiplication**

A = np.array([[1, 2], [3, 4]])

B = np.array([[5, 6], [7, 8]])

print(np.dot(A, B))

**# 44. Index error exception**

try:

a = [1, 2, 3]

print(a[5])

except IndexError:

print("Index out of range")

**# 45. Create dictionary from list**

pairs = [("a", 1), ("b", 2)]

print(dict(pairs))

**# 46. Check palindrome with loop**

s = input()

rev = ""

for ch in s:

rev = ch + rev

print("Palindrome" if rev == s else "Not")

**# 47. Sum digits of number**

n = int(input())

s = 0

while n:

s += n % 10

n //= 10

print(s)

**# 48. Simple calculator**

a = int(input())

b = int(input())

op = input()

if op == '+': print(a + b)

elif op == '-': print(a - b)

elif op == '\*': print(a \* b)

elif op == '/': print(a / b)

**# 49. Reverse list without reverse()**

l = [1, 2, 3]

print(l[::-1])

**# 50. Count frequency of chars in string**

s = "hello"

print(dict(Counter(s)))