



# 200 Python Core Practice Questions

## 1. Basics & Syntax (15 Qs)

1. Print "Hello, Python!" without using `print()`.
2. Swap two numbers without using a third variable.
3. Take input of two numbers and print their sum, difference, product, and quotient.
4. Find the ASCII value of a given character.
5. Convert temperature from Celsius to Fahrenheit and vice versa.
6. Check if a number is even or odd.
7. Find the largest of three numbers.
8. Calculate the area and circumference of a circle.
9. Take user input as a string and print its reverse.
10. Count vowels and consonants in a string.
11. Check if a year is a leap year.
12. Find the square root of a number without using `math.sqrt`.
13. Convert decimal number to binary, octal, and hexadecimal.
14. Take three sides of a triangle and check if they form a valid triangle.
15. Find the factorial of a number using loops.

---

## 2. Conditional & Loops (20 Qs)

16. Print multiplication table of any number.
17. Print Fibonacci series up to N terms.

18. Check if a number is prime.
19. Find all prime numbers in a given range.
20. Print sum of digits of a number.
21. Reverse a number (without converting to string).
22. Check if a number is palindrome.
23. Print Armstrong numbers in a given range.
24. Calculate power of a number using loops.
25. Find GCD and LCM of two numbers.
26. Print pattern:

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

27. Print pattern:

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

28. Print Pascal's triangle.
29. Print all factors of a number.
30. Check if a number is strong number (sum of factorial of digits = number).
31. Find sum of series:  $1 + 1/2 + 1/3 + \dots + 1/n$ .
32. Check Harshad number (divisible by sum of its digits).
33. Generate prime factors of a number.

- 34. Print Floyd's triangle.
  - 35. Find HCF using Euclid's algorithm.
- 

### 3. Strings (25 Qs)

- 36. Check if a string is palindrome.
- 37. Remove all duplicate characters from a string.
- 38. Count words in a string.
- 39. Find frequency of each character in a string.
- 40. Remove vowels from a string.
- 41. Check if two strings are anagrams.
- 42. Find the longest word in a sentence.
- 43. Replace spaces with hyphens.
- 44. Convert string into title case.
- 45. Count uppercase and lowercase letters.
- 46. Remove punctuation from a string.
- 47. Find substring without using `in` or `find()`.
- 48. Sort characters of string alphabetically.
- 49. Find first non-repeating character.
- 50. Find longest common prefix of two strings.
- 51. Check if string contains only digits.
- 52. Convert Roman numeral to integer.
- 53. Check if two strings are rotations of each other.
- 54. Remove all whitespaces without using `replace()`.

- 55. Find all substrings of a string.
  - 56. Find duplicate words in a sentence.
  - 57. Print frequency of words in a paragraph.
  - 58. Check if string is pangram.
  - 59. Find smallest and largest word in string.
  - 60. Reverse each word in a sentence.
- 

## 4. Lists (25 Qs)

- 61. Find second largest number in a list.
- 62. Remove duplicates from list.
- 63. Find common elements between two lists.
- 64. Find difference of two lists.
- 65. Reverse a list without using reverse().
- 66. Rotate list by k elements.
- 67. Find missing number in a list of 1 to n.
- 68. Find all pairs with given sum.
- 69. Flatten a nested list.
- 70. Find largest and smallest element in list.
- 71. Sort list without using sort().
- 72. Count frequency of elements.
- 73. Find sublist with maximum sum.
- 74. Find intersection of multiple lists.
- 75. Merge two sorted lists into one.

- 76. Split list into even and odd numbers.
  - 77. Find duplicates in list.
  - 78. Find kth largest element.
  - 79. Find product of all elements in list.
  - 80. Partition list into positive and negative numbers.
  - 81. Generate power set of a list.
  - 82. Find all permutations of a list.
  - 83. Find all combinations of a list.
  - 84. Find triplets that sum to zero.
  - 85. Replace every element with product of other elements.
- 

## **5. Tuples, Sets, Dictionaries (20 Qs)**

- 86. Convert list into tuple and vice versa.
- 87. Find common elements using sets.
- 88. Check subset and superset.
- 89. Remove duplicates using set.
- 90. Find symmetric difference of two sets.
- 91. Merge two dictionaries.
- 92. Find key with maximum value in dict.
- 93. Sort dictionary by keys.
- 94. Sort dictionary by values.
- 95. Check if two dictionaries are equal.
- 96. Invert dictionary (swap keys and values).

- 97. Count frequency of words using dict.
  - 98. Find all keys with same value.
  - 99. Create dictionary of squares (1–n).
  - 100. Check if key exists in dictionary.
  - 101. Remove key from dictionary safely.
  - 102. Find min and max value in dictionary.
  - 103. Find most frequent element using dictionary.
  - 104. Group elements by length into dict.
  - 105. Merge multiple dictionaries into one.
- 

## **6. Functions (20 Qs)**

- 106. Write function for factorial.
- 107. Function for Fibonacci.
- 108. Function for checking prime.
- 109. Function for reversing a number.
- 110. Function for palindrome string check.
- 111. Function for Armstrong number check.
- 112. Function for GCD & LCM.
- 113. Function for sorting list without built-in sort.
- 114. Recursive function for factorial.
- 115. Recursive function for Fibonacci.
- 116. Function to calculate  $nCr$  and  $nPr$ .
- 117. Function to check strong number.

- 118. Function to generate Pascal triangle.
  - 119. Function to return unique elements in list.
  - 120. Function to check leap year.
  - 121. Function to find digital root of a number.
  - 122. Function to find sum of digits until single digit.
  - 123. Function to print prime factors.
  - 124. Function to count words in sentence.
  - 125. Function to calculate power without `**`.
- 

## 7. File Handling (15 Qs)

- 126. Read a text file and print contents.
- 127. Count number of lines, words, characters.
- 128. Find longest line in file.
- 129. Copy contents of one file to another.
- 130. Merge two files into one.
- 131. Append data into file.
- 132. Find frequency of words in file.
- 133. Remove all blank lines from file.
- 134. Write list into file line by line.
- 135. Read CSV file without using `pandas`.
- 136. Reverse contents of a file.
- 137. Count number of vowels in file.
- 138. Find most repeated word in file.

- 139. Check if file is empty.
  - 140. Compare two files line by line.
- 

## 8. OOP (20 Qs)

- 141. Create a class `Car` with attributes and methods.
- 142. Implement inheritance with `Animal` → `Dog`.
- 143. Implement multiple inheritance.
- 144. Demonstrate method overriding.
- 145. Demonstrate class method and static method.
- 146. Implement encapsulation (private variables).
- 147. Overload `+` operator for custom class.
- 148. Implement polymorphism with same method name.
- 149. Create a `BankAccount` class with deposit/withdraw.
- 150. Create `Student` class with marks and average.
- 151. Create `Library` class with book issue/return.
- 152. Create `Employee` class with salary hike method.
- 153. Implement `__str__` and `__repr__`.
- 154. Create singleton class.
- 155. Demonstrate abstract class.
- 156. Demonstrate multiple constructors.
- 157. Create simple calculator using class.
- 158. Create a shape class with area and perimeter.
- 159. Create Inventory management class.



160. Create Stack class with push/pop.

---

## **9. Exception Handling & Modules (10 Qs)**

- 161. Handle divide by zero exception.
  - 162. Handle file not found error.
  - 163. Raise custom exception.
  - 164. Demonstrate try-except-finally.
  - 165. Demonstrate multiple exception handling.
  - 166. Write module for math operations.
  - 167. Import module and use in program.
  - 168. Write program to log errors in file.
  - 169. Demonstrate assertion.
  - 170. Demonstrate exception chaining.
- 

## **10. Advanced (30 Qs)**

- 171. Implement binary search.
- 172. Implement bubble sort.
- 173. Implement selection sort.
- 174. Implement insertion sort.
- 175. Implement merge sort.
- 176. Implement quick sort.
- 177. Find median of list without using statistics.

178. Implement stack using list.
179. Implement queue using list.
180. Implement circular queue.
181. Implement priority queue.
182. Implement linked list.
183. Implement doubly linked list.
184. Implement binary tree traversal (inorder).
185. Implement BFS traversal.
186. Implement DFS traversal.
187. Count leaf nodes in binary tree.
188. Find height of binary tree.
189. Find nth Fibonacci using dynamic programming.
190. Implement Tower of Hanoi.
191. Implement knapsack problem (0/1).
192. Implement  $nCr$  using DP.
193. Solve Josephus problem.
194. Solve N-Queens problem.
195. Solve Sudoku solver (backtracking).
196. Generate all balanced parentheses.
197. Implement LRU cache.
198. Implement word count using collections.Counter.
199. Implement decorator for timing a function.
200. Create generator function for Fibonacci.