Basic Coding Questions

1. Write a Python program to check if a variable is of type integer.
   Answer: isinstance(x, int)
   Explanation: isinstance() checks if a variable is of a certain type.
2. Create a string and convert it to uppercase and lowercase.
   Answer: s = "hello"; print(s.upper()); print(s.lower())
   Explanation: upper() converts a string to uppercase, lower() converts to lowercase.
3. Write a Python program to check if a variable is of type float.
   Answer: isinstance(x, float)
   Explanation: isinstance() checks if a variable is of a certain type.
4. Create a list and append, insert, and remove elements.
   Answer: my_list = [1, 2, 3]; my_list.append(4); my_list.insert(0, 0); my_list.remove(2)
   Explanation: append() adds an element to the end, insert() adds at a specific position, remove() removes the first occurrence.
5. Write a Python program to create a tuple and extract elements.
   Answer: my_tuple = (1, 2, 3); print(my_tuple[0])
   Explanation: Tuples are immutable, [] extracts an element.
6. Create a dictionary and add, remove, and access key-value pairs.
   Answer: my_dict = {"a": 1}; my_dict["b"] = 2; del my_dict["a"]; print(my_dict["b"])
   Explanation: [] accesses a value, del removes a key-value pair.
7. Write a Python program to check if a number is even or odd using if-else.
   Answer: x = 5; if x % 2 == 0: print("even") else: print("odd")
   Explanation: % is the modulus operator, if-else checks conditions.
8. Create a program to check if a character is vowel or consonant using if-elif-else.
   Answer: c = "a"; if c in "aeiou": print("vowel") elif c.isalpha(): print("consonant")
   Explanation: in checks if a character is in a string, isalpha() checks if a character is a letter.
9. Write a Python program to check if a number is positive, negative, or zero using if-elif-else.
   Answer: x = 5; if x > 0: print("positive") elif x < 0: print("negative") else: print("zero")
   Explanation: if-elif-else checks conditions.
10. Write a Python program to print numbers from 1 to 10 using a while loop.
    Answer: i = 1; while i <= 10: print(i); i += 1
11. Create a program to calculate the sum of numbers from 1 to n using a while loop.
    Answer: n = 10; sum = 0; i = 1; while i <= n: sum += i; i += 1; print(sum)
    Explanation: while loops until a condition is met, += adds to a variable.
12. Write a Python program to print even numbers from 1 to 20 using a while loop.
    Answer: i = 1; while i <= 20: if i % 2 == 0: print(i); i += 1
    Explanation: while loops until a condition is met, % is the modulus operator.
13. Write a Python program to use break to exit a loop when a condition is met.
    Answer: i = 1; while i <= 10: if i == 5: break; print(i); i += 1
    Explanation: break exits a loop when a condition is met.
14. Create a program to use continue to skip a iteration when a condition is met.
    Answer: i = 1; while i <= 10: if i == 5: continue; print(i); i += 1
    Explanation: continue skips to the next iteration when a condition is met.
15. Write a Python program to use pass to do nothing when a condition is met.
    Answer: i = 1; while i <= 10: if i == 5: pass; print(i); i += 1
    Explanation: pass does nothing when a condition is met.

16. Write a Python program to convert a list of strings to a list of integers.
    Answer: my_list = ["1", "2", "3"]; my_list = [int(x) for x in my_list]
    Explanation: List comprehension converts each element to an integer.
17. Create a program to check if a variable is of type complex.
    Answer: isinstance(x, complex)
    Explanation: isinstance() checks if a variable is of a certain type.
18. Write a Python program to convert a tuple to a list.
    Answer: my_tuple = (1, 2, 3); my_list = list(my_tuple)
    Explanation: list() converts a tuple to a list.
19. Create a program to merge two dictionaries.
    Answer: dict1 = {"a": 1}; dict2 = {"b": 2}; dict1.update(dict2)
    Explanation: update() merges two dictionaries.
20. Write a Python program to create a set from a list.
    Answer: my_list = [1, 2, 2, 3]; my_set = set(my_list)
    Explanation: set() creates a set from a list, removing duplicates.
21. Create a program to find the intersection of two sets.
    Answer: set1 = {1, 2, 3}; set2 = {2, 3, 4}; print(set1 & set2)
    Explanation: & operator finds the intersection of two sets.
22. Write a Python program to check if a number is divisible by 3 and 5.
    Answer: x = 15; if x % 3 == 0 and x % 5 == 0: print("divisible")
    Explanation: % is the modulus operator, and checks multiple conditions.
23. Create a program to check if a character is uppercase or lowercase.
    Answer: c = "A"; if c.isupper(): print("uppercase") elif c.islower(): print("lowercase")
    Explanation: isupper() and islower() check if a character is uppercase or lowercase.
24. Write a Python program to check if a string contains a substring.
    Answer: s = "hello world"; if "world" in s: print("contains")
    Explanation: in operator checks if a substring is in a string.
25. Write a Python program to print numbers from 10 to 1 using a while loop.
    Answer: i = 10; while i >= 1: print(i); i -= 1
    Explanation: while loops until a condition is met, -= decrements a variable.
26. Create a program to calculate the sum of squares of numbers from 1 to n.
    Answer: n = 10; sum = 0; i = 1; while i <= n: sum += i ** 2; i += 1; print(sum)
    Explanation: while loops until a condition is met, ** is the exponentiation operator.
27. Write a Python program to print odd numbers from 1 to 20 using a while loop.
    Answer: i = 1; while i <= 20: if i % 2 != 0: print(i); i += 1
    Explanation: while loops until a condition is met, % is the modulus operator.
28. Write a Python program to use break to exit a loop when a sum exceeds 100.
    Answer: sum = 0; i = 1; while i <= 10: sum += i; if sum > 100: break; i += 1
    Explanation: break exits a loop when a condition is met.
29. Create a program to use continue to skip a iteration when a number is even.
    Answer: i = 1; while i <= 10: if i % 2 == 0: continue; print(i); i += 1
    Explanation: continue skips to the next iteration when a condition is met.
30. Write a Python program to use pass to do nothing when a condition is met.
    Answer: i = 1; while i <= 10: if i == 5: pass; print(i); i += 1
    Explanation: pass does nothing when a condition is met.
31. Write a Python program to create a list of squares of numbers from 1 to 10.
    Answer: squares = [i ** 2 for i in range(1, 11)]
    Explanation: List comprehension creates a list of squares.

32. Create a program to create a list of cubes of numbers from 1 to 10.
    Answer: cubes = [i ** 3 for i in range(1, 11)]
    Explanation: List comprehension creates a list of cubes.
33. Write a Python program to create a list of prime numbers from 1 to 20.
    Answer: primes = [i for i in range(2, 21) if all(i % j != 0 for j in range(2, i))]
    Explanation: List comprehension creates a list of prime numbers.
34. Write a Python program to create a function to calculate the area of a rectangle.
    Answer: def area(length, width): return length * width
    Explanation: Function takes length and width as arguments and returns the area.
35. Create a program to create a function to check if a number is prime.
    Answer: def is_prime(n): return all(n % i != 0 for i in range(2, n))
    Explanation: Function takes a number as an argument and returns True if prime, False otherwise.
36. Write a Python program to check if a string is palindrome or not.
    Answer: def is_palindrome(s): return s == s[::-1]
    Explanation: Function takes a string as an argument and returns True if palindrome, False otherwise.
37. Create a program to find the maximum number in a list.
    Answer: numbers = [1, 2, 3]; max_num = max(numbers)
    Explanation: max() function finds the maximum number in a list.
38. Write a Python program to find the factorial of a number.
    Answer: def factorial(n): return 1 if n == 0 else n * factorial(n-1)
    Explanation: Recursive function calculates the factorial of a number.
39. Create a program to check if a number is prime or not.
    Answer: def is_prime(n): return all(n % i != 0 for i in range(2, n))
    Explanation: Function takes a number as an argument and returns True if prime, False otherwise.
40. Write a Python program to print the Fibonacci sequence up to n.
    Answer: def fibonacci(n): a, b = 0, 1; for _ in range(n): print(a); a, b = b, a + b
    Explanation: Function prints the Fibonacci sequence up to n.