**20 interview questions for Operators**

1. What is the difference between the ‘and’ and ‘or’ operators in Python?

The ‘and’ operator returns True if both conditions are true, while the ‘or’ operator returns True if at least one condition is true.

2. What is the modulo operator in Python, and how is it used?

The modulo operator (%) returns the remainder of a division operation.

3. What is the floor division operator in Python?

The floor division operator (//) returns the quotient of a division operation, rounded down to the nearest integer.

4. What is the difference between the ‘==’ and ‘is’ operators in Python?

The ‘==’ operator tests for equality between two objects, while the ‘is’ operator tests whether two objects are the same object in memory.

5. What are the bitwise operators in Python, and how are they used?

Bitwise operators perform operations on the binary representations of integers. These operators include ‘&’ (bitwise and), ‘|’ (bitwise or), ‘^’ (bitwise xor), ‘~’ (bitwise not), ‘<<’ (bitwise left shift), and ‘>>’ (bitwise right shift).

6. How do you perform string concatenation in Python?

String concatenation can be performed using the ‘+’ operator.

7. What is a lambda function in Python, and how is it used?

A lambda function is a small, anonymous function that can be defined on the fly. It is typically used for short, one-time-use functions.

8. How do you use the ternary operator in Python?

The ternary operator in Python is a shorthand way to write an if-else statement in a single line of code. It has the format: value\_if\_true if condition else value\_if\_false.

9. What is the order of operator precedence in Python?

The order of operator precedence in Python follows the usual mathematical order: parentheses, exponentiation, multiplication and division, addition and subtraction, comparison operators, and logical operators.

10. What is the difference between a shallow copy and a deep copy in Python?

A shallow copy creates a new object that references the same memory as the original object, while a deep copy creates a new object with its own memory space.

11. How do you use the ‘in’ operator in Python?

The ‘in’ operator is used to test whether a value is present in a sequence, such as a list, tuple, or string.

12. What is the difference between a list and a set in Python?

A list is an ordered collection of elements, while a set is an unordered collection of unique elements.

13. What is a generator in Python, and how is it used?

A generator is a special type of function that can generate a sequence of values on-the-fly, without storing them all in memory at once.

14. How do you use the ‘range’ function in Python?

The ‘range’ function is used to generate a sequence of numbers with a specified starting value, ending value, and step size.

15. What is the purpose of the ‘map’ function in Python?

The ‘map’ function is used to apply a function to each element of an iterable object and return the results as a new iterable.

16. How do you use the ‘zip’ function in Python?

The ‘zip’ function is used to combine two or more iterables into a single iterable of tuples, where each tuple contains one element from each of the input iterables.

17. What is a tuple unpacking in Python?

Tuple unpacking is a technique that allows you to assign the elements of a tuple to multiple variables in a single line of code.

18. What is the purpose of the ‘\*args’ and ‘\*\*kwargs’ syntax in Python functions?

The ‘\*args’ syntax is used to pass a variable number of non-keyword arguments to a function, while the ‘\*\*kwargs’ syntax is used to pass a variable number of keyword arguments to a function.

19. What is a decorator in Python, and how is it used?

A decorator is a special type of function that can modify the behavior of another function without changing its source code. It is typically used to add functionality to existing functions, such as logging or caching.

20. What is the purpose of the ‘\_\_init\_\_’ method in Python classes?

The ‘init’ method is a special method that is automatically called when a new object is created from a class. It is used to initialize the object’s attributes and perform any necessary setup operations. iterations without encountering a break statement.