**20 Python Interview Questions**

1. **Differentiate between lists and tuples in Python?**

**Answer:** Lists are mutable, while tuples are immutable. This means that elements of a list can be modified after creation, but elements of a tuple cannot.

1. **Explain the concept of mutability in Python with an example.**

**Answer:** Mutability refers to whether an object's state can be modified after it is created. For example, lists are mutable, so you can change their elements, but strings are immutable.

1. **How does Python handle dynamic typing?**

**Answer:** Python is dynamically typed, meaning that variable types are assigned during runtime. You don't need to declare the type explicitly; it is inferred based on the assigned value.

1. **What is the difference between `==` and `is` in Python?**

**Answer:** `==` checks for equality of values, while `is` checks for identity (whether two variables refer to the same object in memory).

1. **Explain the use of the `//` operator in Python?**

**Answer:** The `//` operator performs floor division, which returns the largest integer less than or equal to the quotient of the division.

1. **What is the purpose of the `%` operator in Python?**

**Answer:** The `%` operator is the modulus operator, which returns the remainder of the division.

1. **Differentiate between `and` and `&` in Python?**

**Answer:** `and` is a logical operator, whereas `&` is a bitwise operator. `and` performs logical AND, while `&` performs bitwise AND on corresponding bits.

1. **Explain the concept of ShortCircuiting in Python?**

**Answer:** ShortCircuiting occurs when the result of a logical expression is determined without evaluating all parts. For example, in `a and b`, if `a` is false, `b` is not evaluated.

1. **What is the purpose of the `ifelifelse` statement in Python?**

**Answer:** It is used for conditional branching. The `if` statement is followed by one or more `elif` (else if) statements and an optional `else` statement.

1. **Explain the ternary operator in Python?**

**Answer:** The ternary operator (`x if condition else y`) is a shorthand for an `ifelse` statement. If the condition is true, it returns `x`; otherwise, it returns `y`.

1. **How does the `pass` statement work in Python?**

**Answer:** `pass` is a nooperation statement. It serves as a placeholder when a statement is syntactically required but no action is desired.

1. **What is the difference between `==` and `is` in the context of conditional statements?**

**Answer:** In the context of conditional statements, `==` checks for equality of values, and `is` checks for identity.

1. **How can you handle multiple exceptions in a single `except` block?**

**Answer:** You can catch multiple exceptions by specifying them as a tuple. For example: `except (TypeError, ValueError):`.

1. **Explain the difference between `for` and `while` loops in Python?**

**Answer:** `for` loops are used for iterating over a sequence (e.g., a list), while `while` loops repeatedly execute a block of code as long as a condition is true.

1. **How can you exit a loop prematurely in Python?**

**Answer:** The `break` statement is used to exit a loop prematurely.

1. **What is the purpose of the `range()` function in a `for` loop?**

**Answer:** `range()` generates a sequence of numbers used in `for` loops. It can take one, two, or three arguments to specify the start, stop, and step.

1. **Explain the concept of an infinite loop and how to avoid it?**

**Answer:** An infinite loop continues indefinitely. To avoid it, ensure that the loop condition eventually becomes false, or use techniques like `break` or `return` to exit the loop.

1. **How does the `break` statement differ from the `continue` statement in a loop?**

**Answer:** `break` is used to exit the loop entirely, while `continue` is used to skip the rest of the code inside the loop for the current iteration.

1. **Explain the difference between parameters and arguments in a function?**

**Answer:** Parameters are variables in a function definition, while arguments are the values passed to a function during a call.

1. **How can you make a function accept a variable number of arguments?**

**Answer:** You can use `args` to allow a function to accept a variable number of positional arguments.