**Python Interview Questions**

**Data Types**

1. What are the main built-in data types in Python?

- Python has several built-in data types, including `int`, `float`, `str`, `list`, `tuple`, `dict`, and `set`.

2. Explain the difference between mutable and immutable data types in Python.

- Mutable data types can be modified after creation, while immutable data types cannot. For example, lists are mutable, while tuples are immutable.

3. How do you convert a string to an integer in most programming languages?

- To convert a string to an integer in many programming languages, you can use functions like `int()` in Python

4. What does immutable mean and what three types of Python core data types are considered immutable?

- An immutable data type is a type of object which cannot be modified after its creation. Numbers, strings, and tuples in Python fall into this category. Although you cannot modify an immutable object in place, you can always create a new one by running an expression.

5. What is the None Type data type, and what does it represent?

- Python's None is a special data type that represents the absence of a value or a meaningful value. It is used to signify the absence of a value when a variable or expression doesn't have a valid value to assign or return.

**Operators**

6. What is the difference between '==' and 'is' in Python?

- `'=='` is used to compare the values of two objects, while `'is'` is used to check if two objects are the same (i.e., they have the same memory address).

7. What is the purpose of the 'in' operator in Python?

- The `'in'` operator is used to check if a value exists in a sequence (e.g., a list, string, or tuple).

8. How do you perform exponentiation in Python?

- You can use the \*\* operator for exponentiation. For example, to calculate 2 to the power of 3, you would write 2 \*\* 3, which equals 8.

9. How can you compare two strings in Python?

- You can use comparison operators such as '==', '!=', '>', '<', '>=', and '<=' to compare strings in Python. These operators compare strings based on their lexicographical (dictionary) order.

10. How do you use the 'is' operator for identity comparison in Python?

- The 'is' operator is used to check if two variables reference the same object in memory. For example, x is y will return True if both x and y refer to the same object.

**Conditional Statements**

11. Explain the 'if-elif-else' statement in Python.

- The 'if-elif-else' statement allows you to execute different blocks of code based on different conditions. If the 'if' condition is not met, it checks 'elif' conditions, and if none of them are met, it falls back to 'else.'

12. What is the ternary conditional operator in Python, and how is it used?

- The ternary conditional operator (or conditional expression) is a one-liner 'if-else' statement. It's used as follows: `value\_if\_true if condition else value\_if\_false`.

13. Explain the ternary conditional expression in Python.

- The ternary conditional expression, also known as the conditional expression, is a concise way to write an `if-else` statement in a single line. It has the form `value\_if\_true if condition else value\_if\_false`.

**Looping Statements**

14. What is the difference between 'for' and 'while' loops in Python?

- 'for' loops are used for iterating over a sequence (e.g., a list), while 'while' loops are used to repeatedly execute a block of code as long as a condition is true.

15. Explain the 'break' and 'continue' statements in Python.

- 'break' is used to exit the current loop, while 'continue' is used to skip the current iteration and move to the next iteration of the loop.

16. Explain the concept of an infinite loop.

- An infinite loop is a loop that never terminates because its condition always evaluates to true. It can lead to the program running indefinitely and should be avoided.

**Functions**

17. What is a function in Python?

- A function is a reusable block of code that performs a specific task. It takes input, processes it, and returns output.

18. What is the difference between 'parameters' and 'arguments' in Python functions?

- Parameters are the placeholders in the function definition, while arguments are the actual values passed to the function when it's called.

19. How can you define a default argument in a Python function?

- Default arguments are specified in the function definition by assigning a value to the parameter, like this: `def my\_function(param=value)`.

20. Explain the 'return' statement in a Python function.

- The 'return' statement is used to specify the value that a function should return when it's called. It also exits the function, so nothing after it is executed in that call.