Here are 20 Python-specific interview questions along with their answers, covering the topics of Data Types, Operators, Conditional Statements, Looping Statements, and Functions:

Data Types:

1. Question: What is the difference between mutable and immutable objects in Python?

Answer: Mutable objects can be modified after creation (e.g., lists), while immutable objects cannot be changed (e.g., tuples, strings).

2. Question: How do you convert a string to an integer in Python?

Answer: You can use the `int()` function to convert a string to an integer, e.g., `int("123")`.

3. Question: Explain the concept of a Python set. How is it different from a list?

Answer: A set is an unordered collection of unique elements. Unlike lists, sets do not allow duplicate values.

Operators:

4. Question:What is the difference between `==` and `is` operators in Python?

Answer:The `==` operator checks for equality of values, while the `is` operator checks if two variables point to the same object in memory.

5. Question: How can you perform matrix multiplication in Python?

Answer:You can use the `@` operator or the `numpy.dot()` function for matrix multiplication.

6. Question:Explain the purpose of the `//` operator in Python.

Answer: The `//` operator is used for floor division, which returns the largest integer less than or equal to the division result.

Conditional Statements:

7. Question How can you write a nested `if` statement in Python?

Answer: You can include another `if` statement within an existing `if` block to create a nested structure.

8. Question: What is the `elif` statement used for in Python?

Answer: The `elif` statement is short for "else if" and is used to check multiple conditions after an initial `if` statement.

9. Question:Explain the ternary conditional expression in Python.

Answer: The ternary expression `x if condition else y` returns `x` if the condition is true, otherwise it returns `y`.

Looping Statements:

10. Question: How can you iterate over a dictionary's keys and values in Python?

Answer: You can use a `for` loop with the `items()` method of a dictionary, e.g., `for key, value in my_dict.items():`.

11. Question: What is the purpose of the `range()` function in Python loops?

Answer: The `range()` function generates a sequence of numbers that can be used for iteration in loops.

12. Question: How can you create an infinite loop in Python?

Answer: You can create an infinite loop using a `while` loop with a condition that is always true.

Functions:

13. Question: How do you define a function that accepts a variable number of arguments?

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

14. Question: What is the difference between `return` and `print` in a function?

Answer: `return` is used to send a value back from a function, while `print` displays output to the console.

15. Question: Explain the concept of a lambda function in Python.

Answer: A lambda function is a small, anonymous function defined with the `lambda` keyword. It can have any number of arguments but only one expression.

16. Question: How do you call a function from another module in Python?

Answer: You can import the module using the `import` statement and then call the function using the module name, e.g., `module_name.function_name()`.

17. **Question:** What is list comprehension, and how is it used?

Answer: List comprehension is a concise way to create lists in Python. It combines a `for` loop and an expression in a single line.

18. **Question:** How can you handle exceptions using a `try` and `except` block?

Answer: The `try` block contains code that might raise an exception. If an exception is raised, the corresponding `except` block is executed.

19. **Question:** What is the purpose of a Python generator function?

Answer: A generator function returns an iterator that generates values on-the-fly, saving memory and allowing lazy evaluation.

20. **Question:** How can you import functions or variables directly from a module?

Answer: You can use the `from module_name import function_name` syntax to import specific functions or variables directly.