Q1. What is the purpose of the try statement?

ANS:

The try statement allows you to define a block of code to be tested for errors while it is being executed. The catch statement allows you to define a block of code to be executed, if an error occurs in the try block.

Q2. What are the two most popular try statement variations?

ANS:

The Different Try/Except Variations

So far we've used a try / except and even a try / except / except , but this is only two-thirds of the story. There are two other optional segments to a try block: else and finally . Both of these optional blocks will come after the try and the except .

Q3. What is the purpose of the raise statement?

ANS:

The RAISE statement stops normal execution of a PL/SQL block or subprogram and transfers control to an exception handler. RAISE statements can raise predefined exceptions, such as ZERO\_DIVIDE or NO\_DATA\_FOUND , or user-defined exceptions whose names you decide.

Q4. What does the assert statement do, and what other statement is it like?

ANS:

Python's assert statement allows we to write sanity checks in your code. These checks are known as assertions, and you can use them to test if certain assumptions remain true while we are developing your code. If any of your assertions turn false, then you have a bug in your code.

Q5. What is the purpose of the with/as argument, and what other statement is it like?

ANS:

In Python, the with statement replaces a try-catch block with a concise shorthand. More importantly, it ensures closing resources right after processing them. A common example of using the with statement is reading or writing to a file. A function or class that supports the with statement is known as a context manager.