

1. What is exception handling in Python?

A. Exception handling in Python is a mechanism to handle runtime errors, allowing a program to continue executing instead of terminating abruptly. Exceptions occur when an error arises during program execution, such as dividing by zero, accessing an invalid index in a list, or opening a non-existent file. Proper exception handling ensures the program can manage these errors gracefully, maintain stability, and provide meaningful error messages to the user.

2. How to you use the try, except and finally blocks for Handling exceptions in Python?

A. Python provides the try, except, else, and finally blocks to implement exception handling. The try block contains the code that may raise an exception. If an error occurs inside the try block, the program immediately jumps to the except block, where the error can be handled or logged. Multiple except blocks can be used to handle different types of exceptions specifically. The optional else block executes if no exceptions are raised in the try block, which is useful for code that should run only when the try succeeds. The finally block is executed regardless of whether an exception occurred or not, making it ideal for cleanup actions like closing files or releasing resources.