1. try: -> part of a block where we have to keep our suspicious code. Try statement is used to trigger exception. If code in try block causes some run time error, then try will throw an exception. Exception is caught by except block.

2. IOError: if the file can’t be opened

KeyboardInterrupt: when an unrequired key is pressed by the user

ValueError: when built-in function receives a wrong argument

EOFError: if End-Of-File is hit without reading any data

ImportError: if it is unable to find the module

3. raise statement is used to raise an exception. You can define what kind of error to raise.

4. The assert statement lets you test if a condition in your code returns True, if not, the program will raise an AssertionError. It is like raise statement, where we can create our custom exceptions.

x = 0

assert x > 0, "number should be greater than 0"

print("even")

---------------------------------------------------------------------------

AssertionError Traceback (most recent call last)

<ipython-input-1-da6003f7a28a> in <module>

1 x = 0

----> 2 assert x > 0, "number should be greater than 0"

3 print("even")

AssertionError: number should be greater than 0

5. with statement in Python is used in exception handling to make the code cleaner and much more readable. It simplifies the management of common resources like file streams. The with statement itself ensures proper acquisition and release of resources. This allows common try…except…finally usage patterns to be encapsulated for convenient reuse.

when "as" is used after "with", it will reassign the the returned object to a new identifier.