1. Exception Processing is important to find exceptions that causes the runtime error. As runtime errors halt the program execution when exception occurs.

Exception Processing is used in Various Applications of which few examples are:

1. Checking Appropriate use of input in an application
2. Checking for Arithmetic exceptions in mathematical executions
3. Checking File I/O exceptions during File handling

2. If Exceptions are not handled flow of program will be broken during the run time which might lead to an abnormal termination of the program. In short inability of program to handle exceptions will result in crashing of program.

3. Python provides **try** and **except** statements for recovering from an exception in your script.

4. **raise** and **assert** are two methods that can be used to trigger manual exceptions in your script.

* **raise** method triggers an exception if condition provided to it turns out to be True.
* **assert** will let the program to continue execution if condition provided to it turns out to be True else exception will be raised

In [1]:

*# Example of raise*

x **=** 10

**raise** Exception(f'X Value Should not exceed 5 The Provided Value of X is {x}')

*# Example of assert*

**assert**(2**==**4), "2 is not equal to 4"

5. Python Provides **else** and **finally** blocks for specifying actions to be executed at termination time, regardless of whether an exceptions exists or not.