1. **raise** and **assert** are the two latest user-defined exception constraints in Python 3.X.

2. In python, users can define custom exceptions by creating a new class. This exception class has to be derived, either directly or indirectly from built-in **Exception** class. This new exception class like other exceptions can be raised using the **raise** statement with an optional error message.

Example:

**class** ToYoungException(Exception):

**def** \_\_init\_\_(self,msg):

self**.**msg **=** msg

age **=** 12

**if** age **<=** 18: **raise** ToYoungException('To Young For Voting')

3. The **process()** method of LoggerAdapter is where the contextual information is added to the logging output. It passes the message and keyword arguments of the logging call, and it passes back modified versions of these to use in the call to the underlying logger.

Other method that can be used is **exception()**, Logs a messgae with level ERROR on this logger. The arguments are interpreted as for **debug()**. Exception info is added to the logging message.

4. **raise** and **assert** are two methods for specifying the text of an exception object's error message. **raise** statement is used to trigger explicit exception, if certain condition is not as per requirement of programmer. It helps in triggering exception as per need of programmer and logic.

There are few assertions that programmer always want to be True to avoid code failure. This type of requirement is fulfilled by **assert** statement. This statement takes a Boolean Condition output of which if True, further program executes. If output of assert statement is False it raises an **Assertion Error**.

5. String-based Exceptions doesn't inherit from Exceptions. so plain exceptions catch all exceptions and not only system.