Exception Handling in Python :-

- We have explored basic python till now from Set 1 to 4 (Set 1 | Set 2 | Set 3 | Set 4).
- In this article, we will discuss how to handle exceptions in Python using try, except, and finally statements with the help of proper examples.
- Error in Python can be of two types i.e. Syntax errors and Exceptions. Errors are problems in a program due to which the program will stop the execution. On the other hand, exceptions are raised when some internal events occur which change the normal flow of the program.

Different types of exceptions in python:-

In Python, there are several built-in Python exceptions that can be raised when an error occurs during the execution of a program. Here are some of the most common types of exceptions in Python:

- SyntaxError: This exception is raised when the interpreter encounters a syntax error in the code, such as a misspelled keyword, a missing colon, or an unbalanced parenthesis.
- TypeError: This exception is raised when an operation or function is applied to an object of the wrong type, such as adding a string to an integer.
- NameError: This exception is raised when a variable or function name is not found in the current scope.
- IndexError: This exception is raised when an index is out of range for a list, tuple, or other sequence types.
- KeyError: This exception is raised when a key is not found in a dictionary.
- ValueError: This exception is raised when a function or method is called with an invalid argument or input, such as trying to convert a string to an integer when the string does not represent a valid integer.
- AttributeError: This exception is raised when an attribute or method is not found on an object, such as trying to access a non-existent attribute of a class instance.
- IOError: This exception is raised when an I/O operation, such as reading or writing a file, fails due to an input/output error.
- ZeroDivisionError: This exception is raised when an attempt is made to divide a number by zero.

- ImportError: This exception is raised when an import statement fails to find or load a module.

These are just a few examples of the many types of exceptions that can occur in Python. It’s important to handle exceptions properly in your code using try-except blocks or other error-handling techniques, in order to gracefully handle errors and prevent the program from crashing.

Example of Exception Handling :-

```
a = [1, 2, 3]

try:

  print ("Second element = %d" %(a[1]))



  print ("Fourth element = %d" %(a[3]))



except:

  print ("An error occurred")
```

Output :-

```
Second element = 2

An error occurred
```