

Types of Exceptions:-

- Syntax Error:

Syntax errors are invalidities in the code's structure. In other words, Python syntax errors happen when the Python interpreter is unable to recognize the structure of statements in code.

→ When the proper syntax of the language is not followed than a syntax error is thrown.

Example :

```
amount = 10000  
if amount > 2999  
    print("You are eligible to purchase")
```

Output:

SyntaxError: invalid syntax

- Type Error:

TypeError is one among the several std. Python exceptions. TypeError is raised whenever an operation is performed on an incorrect/unsupported object type. for ex. using the + (addition) operator on a string and an integer value will raise a TypeError. Ex.

```
num1 = 1  
num2 = "two"  
print(num1 + num2)
```

Output:

TypeError: must be str, not int

- NameError:

In Python, NameError is raised when the program attempts to access or use a variable that has not been defined or assigned a value. This can happen if the variable is spelled incorrectly,

or if it is accessed before it has been defined.

Ex:

```
num1 = 9
```

```
print(num)
```

Output:

NameError: name 'num' is not defined

- IndexError:

In Python, IndexError occurs when trying to access an element in a list, tuple or any other sequence using an index that is outside the valid range of indices for that sequence. List Index out of Range occur when an item from a list is tried to be accessed that is outside the range of the list.

Ex:

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

```
print(num[4])
```

Output:

IndexError: list index out of range

- KeyError:

KeyError occurs when we try to access a key that is not present. For ex, we have stored subjects taken by students in a dictionary

with their names as a key and subjects as a value and if we want to get the value of a key that doesn't exist then we get a KeyError.

Ex:

```
subjects = { 'Keshav': 'TOC',
             'Manvik': 'DBMS',
             'Vinayak': 'DMS'}
```

```
print(subjects['Eklvaya'])
```

Output:

KeyError: 'Eklvaya'

- Value Error:

Value Error is occurred when an invalid value is assigned to a variable or passed a function while calling it. It also often occurs during unpacking of sequence data types as well as with functions when a return statement is used. Ex:

```
a = 34
```

```
b = "Hello"
```

```
print(float(a)) # works
```

```
print(float(b)) # ValueError
```

Output: 34.0

ValueError: could not convert string to float: 'Hello'

- AttributeError:
The Python `AttributeError` is raised when an invalid attribute reference is made. This can happen if an attribute or function not associated with a datatype is referenced on it. For example, if a method is called on an integer value. For ex:

```
i = 1  
i.append(2)
```

Output:

`AttributeError: 'int' object has no attribute 'append'`

- ZeroDivisionError:

ZeroDivisionError is raised when a program attempts to perform a division operation where the denominator is zero. This situation is mathematically undefined and Python like many other programming languages, raises an exception to signal the error.

Ex:

```
num = 10  
dnr = 0  
result = num/dnr
```

Output:

`ZeroDivisionError: division by zero`

- Indentation Error:

In Python coding, we have to write code that is appropriately formatted with perfect use of gaps to make the code executable. This perfect use of gaps is termed as indentation. If the user does not write code with proper indentation, it generates an Indentation Error. Ex.

```
def hello():
    print("hello")
    hello
```

Output:

```
IndentationError: expected an indented block after function definition line 2
```

- Module Not Found Error:

This Error occurs when you have not installed that module on your system and try to access the module that the Python interpreter cannot locate in its search path or in the Python path, if python cannot find the module you're trying to import in any of the locations, it raises exception.

```
import module
```

Output:

```
ModuleNotFoundError: No module named 'module'
```

• FileExistsError:

`os.mkdir()` method in Python is used to create a directory in Python or create a directory with Python named path with the specified numeric mode. This method raises `FileExistsError` if the directory to be created already exists. Ex:

```
import os  
directory = "hello"  
parent_dir = "/home/User/Documents"  
path = os.path.join(parent_dir, directory)  
os.mkdir(path)  
print("Directory '%s' created." % directory)
```

In the above example Python script uses '`os.mkdir()`' to create a directory named "Hello" in the "/home/User/Documents" path. If the "Hello" directory already exists then,

Output

```
fileexistserror: file exists: '/home/User/Documents/Hello'
```

• FileNotFoundException:

`FileNotFoundException` is an exception in Python that is raised when a program tries to access a file that doesn't exist. This error typically occurs when you attempt to perform file operations like opening, reading, writing, or deleting a file. Python cannot

find the specified file at the provided path or location.
for ex:

```
file1 = open("abc.txt")
```

Output:

```
FileNotFoundException: No such file or directory : 'abc.txt'
```