

User define Exception

- When we use python on real life application . Lets take an example for age . Age can't be enter in negative value .
- What if the user gave negative age as a input .But these types of constraints cannot be applied in the python programs automatically
- To handle this type of exception we use user define exception
- A program automatically terminates after showing which inbuilt exception has occurred while executing the program when it reaches into an undesired state. We can stop it by using user define exception .
- User defined exceptions can be implemented by raising an exception explicitly,

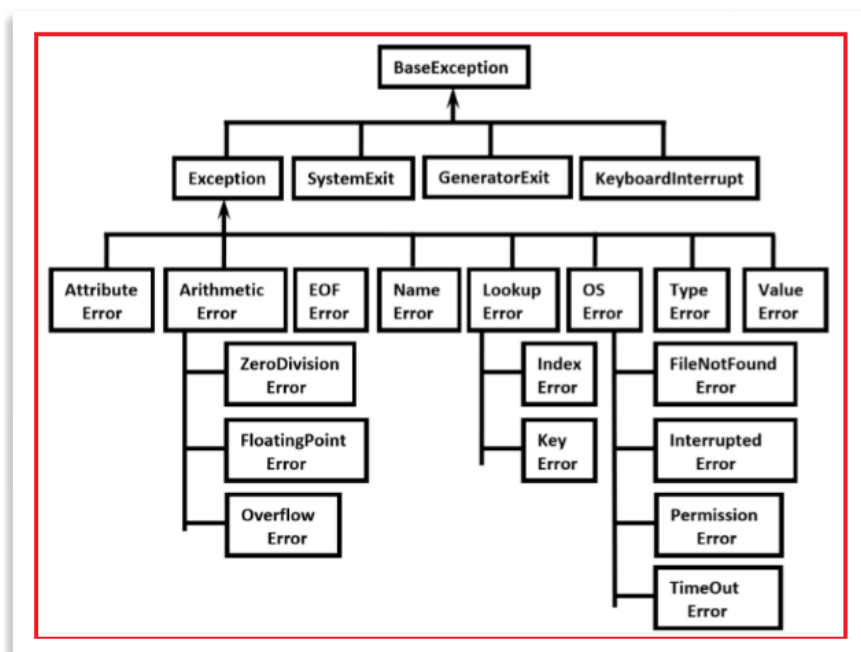
```
class MyError(Exception):  
    pass  
  
try:  
    raise MyError('My Error')  
except MyError as e:  
    print(e)
```

-
- The syntax for raise statement is `raise ExceptionName` . When an error is raised, the code in the except block should handle the exception otherwise it will cause error in the program.
- So, we can see that after using raise statement in try except block, the program gives correct output in both the cases. The diagrams show the built in exception.

We have discussed about , exception , athematic error , zero division etc ...

The built-in exceptions listed below can be generated by the interpreter or built-in functions

User code can raise built-in exceptions. This can be used to test an exception handler



A single try statement can have multiple except statements. This is useful when the try block contains statements that may throw different types of exceptions.

```
class MyError(Exception):
    def __init__(self, msg):
        self.msg=msg

    def __str__(self):
        return self.msg

try:
    raise MyError('My Error')
except MyError as e:
    print(e)
```