Exception and Error Handling

- Until now error messages haven't been more than mentioned, but if you have tried out the examples you have probably seen some.
- There are (at least) two distinguishable kinds of errors: syntax errors and exceptions.

Exceptions

- Even if a statement or expression is syntactically correct, it may cause an error when an attempt is made to execute it.
- Errors detected during execution are called exceptions and are not unconditionally fatal:
- you will soon learn how to handle them in Python programs.
- Most exceptions are not handled by programs, however, and result in error messages as shown here

In [3]:
```
try:
    a=5
    b='0'
# data level
# code level
# system level
    print('a data type is : ',type(a))
    print('b data type is : ',type(b))
    print(a/b)
except Exception as e:
    print("Exception block error msg: ", e)
```

```
a data type is :  <class 'int'>
b data type is :  <class 'str'>
Exception block error msg:  unsupported operand type(s) for /: 'int' and 'str'
```

Handling Exceptions

- It is possible to write programs that handle selected exceptions.

Try Block

- here we can write our code - main execution code we can write in try block #### Exception Block
- here we can handle whichever excpetion or error thorowing by try block

In [ ]:
```
```

In [4]:
```
a=55
b=0
c=a/b
print(c)
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Input In [4], in <cell line: 3>()
      1 a=55
      2 b=0
----> 3 c=a/b
      4 print(c)

rror: division by zero
```

Loading [MathJax]/extensions/Safe.js

In [5]:
```
a = 4
b = 4
try :
    c = a+b
    print('Try Block c value is  : ' ,c)
except Exception as e:

    print("Exception error message: => ",e)
```

```
Try Block c value is  :  8
```

- Multiple Exception handling

In [6]:
```
def divbyzero(x,y):
    try:
        return x/y
    except Exception as e:
        print('Exception raised - Please verify : ',e)
```

In [7]:
```
divbyzero(5,0)
```

```
Exception raised - Please verify :  division by zero
```

In [8]:
```
def divbyzero(x,y):
    try:
        return x/y
    except ZeroDivisionError as e:
        print('ZeroDivisionError and error message :',e)
    except Exception as e:
        print('Others Exception  error message : ',e)
```

In [11]:
```
list_a=[1,2,3,4]
list_a[1]
```

Out[11]:
```
2
```

In [12]:
```
#Calling function with second argument with zero (0)
divbyzero(10,'1')
```

```
Others Exception  error message :  unsupported operand type(s) for /: 'int' and 'str'
```

In [13]:
```
divbyzero(10,0)
```

```
ZeroDivisionError and error message : division by zero
```

In [14]:
```
def int_func(x,y):
    z = 0
    try:
        z = x + y

    except:
        print('Please input only integer or float value')

    return z
```

In [15]:
```
int_func(1,'4')
```

```
Please input only integer or float value
```

Out[15]:
```
0
```

In [16]:
```
a = 33
```

Loading [MathJax]/extensions/Safe.js

```
b = 78
a_list = [1,1,2,3]
try :
    c = a/b
    e = a_list[0]
    print(e)
except ZeroDivisionError:
    print("Handled ZeroDivisionError")

else:
    print('this is else block : ',c)
```

```
1
this is else block :  0.4230769230769231
```

In [17]:

```
a = 44
b = 1
d = [1,2,3,4]
try :
    c = a/b
    e = d[9]
    print('Try Block ',e)
except ZeroDivisionError as e:
    print("Execution is caught ZeroDivisionError : ",e)
except IndexError as e:
    print("Exception is caught is list index out of range : ",e)

else:# only if Try block is True this will be executed.
    print('Else Block : ',c)
    print('Else Block : ',e)
```

```
Exception is caught is list index out of range :  list index out of range
```

In [18]:

```
a = 66
b = 33
d = []
try :

    c = a/b
    e = d[5]
except:
    print('Exception is caught')
else:
    print(c)
```

```
Exception is caught
```

In [19]:

```
a = 44
b = 66
d = []
try :
    c = a/b
    e = d[5]
    print(e)
except Exception as e:
    print("Exception Error Message :  " +str(e))
    # email alert
    # inserting into error table
    # inserting into error log file


else:
```

Loading [MathJax]/extensions/Safe.js

```
    print(c)
print("Good Tracking")
```

```
Exception Error Message :  list index out of range
Good Tracking
```

In [20]:

```
a = 88
b = 0
d = [1,2,3,4,5]
try :
    c = a/b
    e = d[0]
    print('Try Blob list index zero value :',e)

except ZeroDivisionError: # this block will be executed only if Try block is returning F
    print("Handle ZeroDivisionError")
except Exception as e: # this block will be executed only if Try block is returning Fals
  print("others ")

else: # this block will be executed only if Try block is returning True
    print('Else Blob c value ' ,c)

finally: # if Try block is returning True or False
    print("Finally will always exuted if we get any exception or not")
```

```
Handle ZeroDivisionError
Finally will always exuted if we get any exception or not
```

Exception Handling with `ELSE`

In [21]:

```
a = 88
b = 0
d = [1,2,3,4,5,6,6]
try :
    c = a/b
    e = d[5]
    print('inside Try Block : ',c)
except ZeroDivisionError:
    print("Handle ZeroDivisionError")
except IndexError:
    print("Exception is caught is list index out of range")
else:
    print('in else block :' ,c)
finally:
    print("Finally will always exuted if we get any exception or not")
```

```
Handle ZeroDivisionError
Finally will always exuted if we get any exception or not
```

Loading [MathJax]/extensions/Safe.js