• Day 14 – Try & Catch

☐ List down all the error types and check all the errors using a python program for all errors

SyntaxError

In [1]:

```
if a < 3
```

```
  File "<ipython-input-1-3e28e520013d>", line 1
    if a < 3
           ^
SyntaxError: invalid syntax
```

ZeroDivisionError

In [3]:

```
1 / 0
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-3-bc757c3fda29> in <module>
----> 1 1 / 0

ZeroDivisionError: division by zero
```

FileNotFoundError

In [4]:

```
open("imaginary.txt")
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-4-2bedf9c67314> in <module>
----> 1 open("imaginary.txt")

FileNotFoundError: [Errno 2] No such file or directory: 'imaginary.txt'
```

IndexError

In [5]:

```
L1=[1,2,3]
L1[3]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-5-30c160f9e63a> in <module>
      1 L1=[1,2,3]
----> 2 L1[3]

IndexError: list index out of range
```

ModuleNotFoundError

In [6]:

```
import notamodule
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-6-6529f882604c> in <module>
----> 1 import notamodule

ModuleNotFoundError: No module named 'notamodule'
```

KeyError

In [7]:

```
D1={'1':"aa", '2':"bb", '3':"cc"}
D1['4']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-7-5b6257ee76ed> in <module>
      1 D1={'1':"aa", '2':"bb", '3':"cc"}
----> 2 D1['4']

KeyError: '4'
```

TypeError

In [8]:

```
'2'+2
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-8-7e1e50100c0b> in <module>
----> 1 '2'+2

TypeError: can only concatenate str (not "int") to str
```

ValueError

In [9]:

```
int('xyz')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-9-f9171139ba2a> in <module>
----> 1 int('xyz')

ValueError: invalid literal for int() with base 10: 'xyz'
```

NameError

In [10]:

```
age
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-10-d075b0315d53> in <module>
----> 1 age

NameError: name 'age' is not defined
```

☐ Design a simple calculator app with try and except for all use cases

In [14]:

```
def calculator():
    try:
        print('+')
        print('*')
        print('-')
        print('/')
        print('%')
        print('**')
        operation=input("select an operator given above:")
        n_1=int(input("Enter first number: "))
        n_2=int(input("Enter second number: "))
        if operation=='+':
            print(n_1+n_2)
        elif operation=='-':
            print(n_1-n_2)
        elif operation=='*':
            print(n_1*n_2)
        elif operation=='/':
            print(n_1/n_2)
        elif operation=='%':
            print(n_1%n_2)
        elif operation=='**':
            print(n_1**n_2)
        else:
            print("Invalid input")
    except Exception as e:
          print(e)
```

In [15]:

```
calculator()
```

```
+
*
-
/
%
**
select an operator given above:*
Enter first number: 6
Enter second number: j
invalid literal for int() with base 10: 'j'
```

In [16]:

```
calculator()
```

```
+
*
-
/
%
**
select an operator given above:/
Enter first number: 8
Enter second number: 0
division by zero
```

In [17]:

```
calculator()
```

```
+
*
-
/
%
**
select an operator given above:+
Enter first number: 6
Enter second number: 9
15
```

□ print one message if the try block raises a NameError and another for other errors

In [22]:

```
try:
    x=int(input("Enter something:"))
    print(y)
except NameError:
     print("name 'y' is not defined")
except ValueError as ve:
     print(ve)
```

```
Enter something:6
name 'y' is not defined
```

□ When try-except scenario is not required?

When an error occurs, or exception as we call it, Python will normally stop and generate an error message. These exceptions can be handled using the try statement.The else keyword to define a block of code to be executed if no errors were raised.

□ Try getting an input inside the try catch block

In [23]:

```
try:
    num=int(input("Enter a number:"))
except Exception as e:
    print(e)
```

```
Enter a number:seven
invalid literal for int() with base 10: 'seven'
```

In [ ]:

```

```