SyntaxError Vs Exception

```
#SyntaxError: invalid syntax
a=5
b=0
print(a/b))
```

```
     File "<ipython-input-1-d19b5e5e8af2>", line 3
       print(a/b))
                  ^
   SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

```
#Exception:ZeroDivisionError: division by zero
a=5
b=0
print(a/b)
```

```
   ---------------------------------------------------------------------------
   ZeroDivisionError                         Traceback (most recent call last)
   <ipython-input-2-b8c4205a9e4b> in <module>()
         2 a=5
         3 b=0
   ----> 4 print(a/b)

   ZeroDivisionError: division by zero
```

SEARCH STACK OVERFLOW

Standard Exceptions in Python

```
# NameError:It occurs when a name is not found.i.e attempt to access an undeclared variable
x=5
z=x+y
print("Sum =",c)
```

```
   ---------------------------------------------------------------------------
   NameError                                 Traceback (most recent call last)
   <ipython-input-4-bcae356ca8a9> in <module>()
         1 # NameError:It occurs when a name is not found.i.e attempt to access an undeclared variable
         2 x=5
   ----> 3 z=x+y
         4 print("Sum =",c)

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

SEARCH STACK OVERFLOW

```
# ZeroDivisionError: Occurs when a number is divided by zero.
a=5
b=0
```

```
print(a/b)
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-5-7a9ed2818246> in <module>()
      2 a=5
      3 b=0
----> 4 print(a/b)

ZeroDivisionError: division by zero
```

SEARCH STACK OVERFLOW

```
# ValueError: Occurs when an inappropriate value assigned to variable.
a=int(input("Enter a number : "))
b=int(input("Enter a number : "))
print("Sum =",a+b)
```

```
Enter a number : 12
Enter a number : ab
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-6-149857f14fd2> in <module>()
      1 #       ValueError: Occurs when an inappropriate value assigned to variable.
      2 a=int(input("Enter a number : "))
----> 3 b=int(input("Enter a number : "))
      4 print("Sum =",a+b)

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

SEARCH STACK OVERFLOW

```
# IndexError: Occurs when we request for an out-of-range index for sequence.
ls=['c','java','python']
print("list item is :",ls[5])
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-7-2f9476e0e6c6> in <module>()
      1 # IndexError: Occurs when we request for an out-of-range index for sequence.
      2 ls=['c','java','python']
----> 3 print("list item is :",ls[5])

IndexError: list index out of range
```

SEARCH STACK OVERFLOW

```
# KeyError: Occurs when we request for a non-existent dictionary key.
dic={"name":"Madhu","location":"Hyd"}
print("The age is :",dic["age"])
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-8-0b040bedec06> in <module>()
```

```
# FileNotFoundError: Occurs when we request for a non-existent input/output file.
fn=open("exam.py")
print(fn)
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-9-b1a879c1b888> in <module>()
      1 #       IOError: Occurs when we request for a non-existent input/output file.
----> 2 fn=open("exam.py")
      3 print(fn)

FileNotFoundError: [Errno 2] No such file or directory: 'exam.py'
```

SEARCH STACK OVERFLOW

Exception Handling

For Exception handling, python uses following keywords or statements

- try
- except
- else
- finally
- raise
- assert

```
# try - except statement
try:
    a = int(input("Enter a:"))
    b = int(input("Enter b:"))
    c = a/b;
    print("a/b = %d"%c)
except Exception:
    print("can't divide by zero")
#other code:
print("other part of the program")
```

```
Enter a:2
Enter b:3
a/b = 0
other part of the program
```

```
# try - except-else statement:
try:
    a = int(input("Enter a:"))
    b = int(input("Enter b:"))
    c = a/b;
    print("a/b = %d"%c)
except Exception:
```

```
        print("can't divide by zero")
    else:
      print("code for else block")
    #other code:
    print("other part of the program")
```

```
Enter a:23
Enter b:3
a/b = 7
code for else block
other part of the program
```

```
    # finally statement:
    try:
        a = int(input("Enter a:"))
        b = int(input("Enter b:"))
        c = a/b;
        print("a/b = %d"%c)
    except ZeroDivisionError:
        print("can't divide by zero")
    except ValueError:
      print("Enter Numbers only")
    else:
      print("code for else block")
    finally:
      print("Imp code-always executes")
```

```
Enter a:23
Enter b:asd
Enter Numbers only
Imp code-always executes
```

```
    # raise statement:
    try:
        age = int(input("Enter the age : "))
        if age<18:
            raise Exception;
        else:
            print("the age is valid")
    except Exception:
        print("The age is not valid")
```

```
Enter the age : 12
The age is not valid
```

```
    try:
        a = int(input("Enter a : "))
        b = int(input("Enter b : "))
        if b is 0:
            raise ArithmeticError;
        else:
            print("a/b = ",a/b)
    except ValueError:
      print("The values must be numbers")
    except ArithmeticError:
```

```
    print("The value of b can't be 0")
```

```
Enter a : 3
Enter b : d
The values must be numbers
```

```
# assert statement:
x = int(input("Enter x :"))
y = int(input("Enter y :"))
# It uses assert to check for 0
assert y != 0, "Divide by 0 error"
print ("x / y value is : ")
print (x / y)
```

```
Enter x :5
Enter y :0
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
<ipython-input-7-046c0039bb1b> in <module>()
      3 y = int(input("Enter y :"))
      4 # It uses assert to check for 0
----> 5 assert y != 0, "Divide by 0 error"
      6 print ("x / y value is : ")
      7 print (x / y)

AssertionError: Divide by 0 error
```

SEARCH STACK OVERFLOW

```
def avg(marks):
    assert len(marks) != 0,"The List is empty."
    return sum(marks)/len(marks)
marks1 = [67,59,86,75,92]
print("The Average of Marks1:",avg(marks1))
marks2 = []
print("The Average of Marks2:",avg(marks2))
```

```
The Average of Marks1: 75.8
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
<ipython-input-8-6c09686a099b> in <module>()
      5 print("The Average of Marks1:",avg(marks1))
      6 marks2 = []
----> 7 print("The Average of Marks2:",avg(marks2))

<ipython-input-8-6c09686a099b> in avg(marks)
      1 def avg(marks):
----> 2     assert len(marks) != 0,"The List is empty."
      3     return sum(marks)/len(marks)
      4 marks1 = [67,59,86,75,92]
      5 print("The Average of Marks1:",avg(marks1))

AssertionError: The List is empty.
```

SEARCH STACK OVERFLOW

```
# Creating Exceptions
class UnderAge(Exception):
    pass
```

```
    pass
def verify_age(age):
    if int(age) < 18:
        raise UnderAge
    else:
        print('Age: '+str(age))
# main program
try:
  age = int(input("Enter the age : "))
  verify_age(age)
except UnderAge:
  print("UnderAgeException: Less Age")
```

```
Enter the age : 12
UnderAgeException: Less Age
```