Colab Link: https://colab.research.google.com/drive/1HJjNV_MjFC6XlzIaEJvLRCsEaSuvRCbw?usp=sharing

```
print(a)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-bca0e2660b9f> in <module>()
----> 1 print(a)

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

SEARCH STACK OVERFLOW

```
if (5>3):
```

```
  File "<ipython-input-4-2c3c3da165b0>", line 1
    if (5>3):
             ^
SyntaxError: unexpected EOF while parsing
```

SEARCH STACK OVERFLOW

```
import dontknow
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-5-bcd87af70373> in <module>()
----> 1 import dontknow

ModuleNotFoundError: No module named 'dontknow'

---------------------------------------------------------------------------
NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To view examples of installing some common dependencies, click the
"Open Examples" button below.
---------------------------------------------------------------------------
```

RFLOW

Saved successfully!

```
"Data Science".sort()
```

```
    ---------------------------------------------------------------------------
```

```
__builtins__
```

```
    <module 'builtins' (built-in)>
    AttributeError: 'str' object has no attribute 'sort'
```

```
print(dir(__builtins__))
```

```
    ntError', 'FutureWarning', 'GeneratorExit', 'IOError', 'ImportError', 'ImportWa
```

```
temp = dir(__builtins__)


count = 0
for element in temp:
  if "Error" in element:
    print(element)
    count += 1
print(count)
```

```
    ArithmeticError
    AssertionError
    AttributeError
    BlockingIOError
    BrokenPipeError
    BufferError
    ChildProcessError
    ConnectionAbortedError
    ConnectionError
    ConnectionRefusedError
    ConnectionResetError
    EOFError
    EnvironmentError
    FileExistsError
    FileNotFoundError
    FloatingPointError
    IOError
    ImportError
    IndentationError
    IndexError
    InterruptedError
    IsADirectoryError
    KeyError
    LookupError
```

Saved successfully! ×

```
    NotADirectoryError
    NotImplementedError
    OSError
    OverflowError
    PermissionError
    ProcessLookupError
    RecursionError
    ReferenceError
    RuntimeError
    SyntaxError
```

```
SystemError
TabError
TimeoutError
TypeError
UnboundLocalError
UnicodeDecodeError
UnicodeEncodeError
UnicodeError
UnicodeTranslateError
ValueError
ZeroDivisionError
48
```

```
# Exception Handling
```

```
x = 1
y = 0
z = x / y
print(x)
print("Hello World")
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-15-cbb7b8ed8096> in <module>()
      1 x = 1
      2 y = 0
----> 3 z = x / y
      4 print(x)
      5 print("Hello World")

ZeroDivisionError: division by zero
```

SEARCH STACK OVERFLOW

```
# try-except
```

```
a = int(input("Enter the inout first"))
b = int(input("Enter the input second"))
```

```
Enter the inout first5
Enter the input second0
```

Saved successfully! ×

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-18-aae42d317509> in <module>()
----> 1 a/b

ZeroDivisionError: division by zero
```

SEARCH STACK OVERFLOW

```
a = int(input("Enter the inout first"))
b = int(input("Enter the input second"))
try:
  result = a/b
  print(f"Output of a/b is {result}")
except:
  print("There is some error")
```

```
Enter the inout first14
Enter the input second0
There is some error
```

```
lst = [2, 0, "helloo", None]
```

```
for element in lst:
  try:
    print(f"Current element: {element}")
    result = 5/element
    print(f"Result is {result}")
  except Exception as e:
    print("Error occured:", e)
```

```
Current element: 2
Result is 2.5
Current element: 0
Error occured: division by zero
Current element: helloo
Error occured: unsupported operand type(s) for /: 'int' and 'str'
Current element: None
Error occured: unsupported operand type(s) for /: 'int' and 'NoneType'
```

```
for element in lst:
  try:
    print(f"Current element: {element}")
    result = 5/element
    print(f"Result is {result}")
  except ValueError as vs:
    print("Provide a valid integer:", vs)
  except ZeroDivisionError as zs:
    print("Provide a different second number",zs)
  except Exception as e:
    print("Error casused", e)
```

Saved successfully! ×

```
Current element: 2
Result is 2.5
--------------------------------------------------
Current element: 0
Provide a different second number division by zero
--------------------------------------------------
Current element: helloo
Error casused unsupported operand type(s) for /: 'int' and 'str'
--------------------------------------------------
Current element: None
```

```
    Error casused unsupported operand type(s) for /: 'int' and 'NoneType'
    --------------------------------------------------
```

```
try:
  ans= 5/0
  print(ans)
except Exception as e:
  print(e)
finally:
  print("I ran finallly")
```

```
    division by zero
    I ran finallly
```

```
try:
  ans= 5/0
  print(ans)
except Exception as e:
  print(e)
print("I ran finally")
```

```
    division by zero
    I ran finally
```

```
# Raise Exceptions
```

```
try:
  name = input("Enter a name ")
  if len(name) < 3:
    raise Exception("Name cannot be less than 3 characters")
  print("Hello", name)
except Exception as e:
  print("Error occured", e)
print("Some 100 lines after this")
```

```
    Enter a name It
    Error occured Name cannot be less than 3 characters
    Some 100 lines after this
```

```
def foo():
```

Saved successfully! ×

```
    print(ans)
  except Exception as e:
    return e
  print("I ran finally")
```

```
def foo2():
  try:
    ans= 5/0
    print(ans)
  except Exception as e:
    return e
  finally:
    print("I ran finally")
```

```
foo()
```

```
ZeroDivisionError('division by zero')
```

```
foo2()
```

```
I ran finally
ZeroDivisionError('division by zero')
```

```
#class MyCustomError(Exception)
```

```
# How to create custom immutable objects
# How can we make our custom Error and python interpreter will be able to categoris
```

Saved successfully! ✕ completed at 21:57 ● ✕