syntax errors

Syntax error occurs when the code structure voilates the language's syntax rules. These errors are detected by python interpreter during parsing phase

In [1]:
```
print('Hello World)
```

```
  Cell In[1], line 1
    print('Hello World)
          ^
SyntaxError: unterminated string literal (detected at line 1)
```

Runtime errors

Runtime error are also known as exceptions , occur during the execution of a program when something unexpected happens that disrupt the normal flow of the program.

In [2]:
```
n1=int(input('enter a number'))
n2=int(input('enter a number'))
print(f'division of {n1} and {n2} is {n1/n2}')
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[2], line 3
      1 n1=int(input('enter a number'))
      2 n2=int(input('enter a number'))
----> 3 print(f'division of {n1} and {n2} is {n1/n2}')

ZeroDivisionError: division by zero
```

Name Error

A name erroe occur when a variable or function is referenced before

In [3]:
```
vipul
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[3], line 1
----> 1 vipul

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

Type Error

A type error when an operation is performed on an object of an inappropriate type

In [18]:
```
a='10'
```

```
b=10
a+b
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[18], line 3
      1 a='10'
      2 b=10
----> 3 a+b

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

index error

an index error occurs when trying to access an index that is outside the range of a sequence

In [7]:
```
l1=['vipul','anshu','ankit'] # right
l1[2]
```

Out[7]:
```
'ankit'
```

In [8]:
```
l63=['vipul','anshu','ankit'] # wrong or error
l63[3]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[8], line 2
      1 l63=['vipul','anshu','ankit'] # wrong or error
----> 2 l63[3]

IndexError: list index out of range
```

Key Error

A key error occurs when trying to access a key in a dictionary that doesn't exists

In [19]:
```
dict1={'name':'Vipul','class':12}
dict1['marks']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[19], line 2
      1 dict1={'name':'Vipul','class':12}
----> 2 dict1['marks']

KeyError: 'marks'
```

Attribute error

An attribute error occurs when trying to access an attribute or method that does't exist fir an object.

In [9]:
```
str1='VIPUL'
str1.lower() # str--> lower,upper,swapcase,replace,etc
l1.lower() # list--> append,insert,remove,sort,pop,clear,index
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[9], line 3
      1 str1='VIPUL'
      2 str1.lower() # str--> lower,upper,swapcase,replace,etc
----> 3 l1.lower()

AttributeError: 'list' object has no attribute 'lower'
```

In [11]:
```
# table
n=int(input('enter a number'))

for i in range(1,11):
    print(f'{n} x {i} = {n*i}')
```

```
45 x 1 = 45
45 x 2 = 90
45 x 3 = 135
45 x 4 = 180
45 x 5 = 225
45 x 6 = 270
45 x 7 = 315
45 x 8 = 360
45 x 9 = 405
45 x 10 = 450
```

In [12]:
```
# try and except blocks
try:
    n=int(input('enter a number'))

    for i in range(1,11):
        print(f'{n} x {i} = {n*i}')
except:
    print('kuch to gadbad hai daya')
```

```
kuch to gadbad hai daya
```

In [14]:
```
try:
    n1=int(input('enter a number'))
    n2=int(input('enter a number'))
    print(f'division of {n1} and {n2} is {n1/n2}')
except ZeroDivisionError:
    print('denominator could not be zero')
```

```
```

```
denominator could not be zero
```

In [15]:
```
try:
    n1=int(input('enter a number'))
    n2=int(input('enter a number'))
    print(f'division of {n1} and {n2} is {n1/n2}')
except ZeroDivisionError:
    print('denominator could not be zero')
except ValueError:
    print('please add valid integers only')
```

```
please add valid integers only
```

In [16]:
```
# exception as e
try:
    n1=int(input('enter a number'))
    n2=int(input('enter a number'))
    print(f'division of {n1} and {n2} is {n1/n2}')
except Exception as e:
    print(e)
```

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

In [17]:
```
# finally
try:
    n1=int(input('enter a number'))
    n2=int(input('enter a number'))
    print(f'division of {n1} and {n2} is {n1/n2}')
except Exception as e:
    print(e)
finally:
    print('end of code')
```

```
division of 10 and 20 is 0.5
end of code
```

In [ ]:
```
```