Q1. What is an exeption in python? Write the diffrence between exception and syntax errors.

In [1]:
```
# ans:-An exception is an event, which occurs during the execution of a program that disrupts
#      the normal flow of the program's instructions
#     Diffrence between Exception and Syntax errors
#    A syntax error is an issue in a program that prevents the program from completing its task.
#    In comparison, an exception is a condition that interrupts the normal flow of the program
```

Q2. What happens when an exception is not handled? Explain with an example

In [1]:
```
#  Ans:-the runtime system will abort the program (i.e. crash) and an exception message will
#        print to the console
```

Q3. Which Python statements are used to catch and handle exceptions? Explain with an example

In [2]:
```
# Ans:-The try and except block in Python is used to catch and handle exceptions.
#      Python executes code following the try statement as a “normal” part of the program.
#      The code that follows the except statement is the program's response to any exceptions in the preceding try clause.
```

Q4. Explain with an example:

In [6]:
```
# Ans:-

# a.TRY and ELSE :-You can use the else keyword to define a block of code to be executed if no errors were raised
```

In [7]:
```
try:
  print("Hello")
except:
  print("Something went wrong")
else:
  print("Nothing went wrong")
```

```
Hello
Nothing went wrong
```

In [8]:
```
# b.finally :-The finally block will always be executed, no matter if the try block raises an error or not:
```

In [10]:
```
x=5
try:
  x > 3
except:
  print("Something went wrong")
else:
  print("Nothing went wrong")
finally:
  print("The try...except block is finished")
```

```
Nothing went wrong
The try...except block is finished
```

In [11]:
```
# c. raise:-Raise an error and stop the program if x is lower than 0:
```

In [12]:
```
x = -1

if x < 0:
  raise Exception("Sorry, no numbers below zero")
```

```
---------------------------------------------------------------------------
Exception                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_1256\3386925827.py in <module>
      2
      3 if x < 0:
----> 4   raise Exception("Sorry, no numbers below zero")

Exception: Sorry, no numbers below zero
```

Q5. What are Custom Exceptions in python? Why do we need Custom Exceptions? Explain with an example

In [13]:
```
# Ans:-To create a custom exception class, you define a class that inherits from the built-in Exception class or one of its subclasses such as Value
# Need of Custom Exceptions:=To catch and provide specific treatment to a subset of existing Java exceptions.
```

In [14]:
```
class InvalidAgeException(Exception):
    "Raised when the input value is less than 18"
    pass


number = 18

try:
    input_num = int(input("Enter a number: "))
    if input_num < number:
        raise InvalidAgeException
    else:
        print("Eligible to Vote")

except InvalidAgeException:
    print("Exception occurred: Invalid Age")
```

```
Enter a number: 25
Eligible to Vote
```

Q6. Create a custom exception class. Use this class to handle an exception.

In [15]:
```
class FahrenheitError(Exception):
    min_f = 32
    max_f = 212

    def __init__(self, f, *args):
        super().__init__(args)
        self.f = f

    def __str__(self):
        return f'The {self.f} is not in a valid range {self.min_f, self.max_f}'
```

In [ ]:
```

```