



What is Exception?

- ▶ An exception is an event, which occurs during the execution of a program, that **disrupts** the normal flow of the program's instructions.
- ▶ In general, when a Python script encounters a situation that it can't cope with, it raises an exception.
- ▶ An exception is a Python object that represents an error.
- ▶ When a Python script raises an exception, it must handle the exception immediately - otherwise the script would **terminate** and return to the command line.



Handling an exception:

- ▶ If you have some *suspicious* code that may raise an exception, you can defend your program by placing the suspicious code in a **try:** block.
- ▶ After the try: block, include an **except:** statement, followed by a block of code which handles the problem as elegantly as possible.

```
try:
    You do your operations here;
    .....
except Exception I:
    If ExceptionI occurs, then execute this block.
except Exception II:
    If ExceptionII occurs, then execute this block.
    .....
else:
    If there is no exception then execute this block.
```



Tips

- ▶ A single try statement can have **multiple** except statements. This is useful when the try block contains statements that may throw **different types** of exceptions.
- ▶ You can also provide a generic **except** clause, which handles any exception.
- ▶ After the except clause(s), you can include an **else-clause**. The code in the else-block executes if the code in the try: block does not raise an exception.
 - ▶ The else-block is a good place for code that does not need the try: block's protection. (e.g. closing a file)



Raising Exceptions

- ▶ You can create an object of an appropriate exception class and raise the object, using the following syntax:

```
raise ValueError("Error message goes here")
```

- ▶ e.g.

```
if age < 0:  
    raise ValueError()  
  
if age < 0:  
    raise ValueError("Invalid age!")
```



Tips

- ▶ Q1 – Q2:
 - ▶ TypeError or ValueError or (TypeError, ValueError)
- ▶ Q3 – Q8: List Processing
 - ▶ TypeError and/or IndexError
 - ▶ Hint: You may need to use try-except **within** a loop so that the loop performing calculations can continue even if an error occurs
- ▶ Q9 – Q10: Dictionary Processing
 - ▶ TypeError, KeyError, ValueError
- ▶ Q11: File Reading and processing
 - ▶ TypeError, FileNotFoundError, ValueError

```
for num in numbers:
    try:
        ...
    except TypeError:
        pass
```