19.4. Standard Exceptions

Most of the standard *exceptions* built into Python are listed below. They are organized into related groups based on the types of issues they deal with.

Language Exceptions	Description
StandardError	Base class for all built-in exceptions except StopIteration and SystemExit.
ImportError	Raised when an import statement fails.
SyntaxError	Raised when there is an error in Python syntax.
IndentationError	Raised when indentation is not specified properly.
NameError	Raised when an identifier is not found in the local or global namespace.
UnboundLocalError	Raised when trying to access a local variable in a function or method but no value has been assigned to it.
TypeError	Raised when an operation or function is attempted that is invalid for the specified data type.
LookupError	Base class for all lookup errors.
IndexError	Raised when an index is not found in a sequence.
KeyError	Raised when the specified key is not found in the dictionary.
ValueError	Raised when the built-in function for a data type has the valid type of arguments, but the arguments have invalid values specified.
RuntimeError	Raised when a generated error does not fall into any category.
MemoryError	Raised when a operation runs out of memory.
RecursionError	Raised when the maximum recursion depth has been exceeded.
SystemError	Raised when the interpreter finds an internal problem, but when this error is encountered the Python interpreter does not exit.

Math Exceptions	Description
ArithmeticError	Base class for all errors that occur for numeric calculation. You know a math error occurred, but you don't know the specific error.
OverflowError	Raised when a calculation exceeds maximum limit for a numeric type.
FloatingPointError	Raised when a floating point calculation fails.
ZeroDivisonError	Raised when division or modulo by zero takes place for all numeric types.

I/O Exceptions	Description
FileNotFoundError	Raised when a file or directory is requested but doesn't exist.
IOError	Raised when an input/ output operation fails, such as the print statement or the open() function when trying to open a file that does not exist. Also raised for operating system-related errors.
PermissionError	Raised when trying to run an operation without the adequate access rights.
EOFError	Raised when there is no input from either the raw_input() or input() function and the end of file is reached.
KeyboardInterrupt	Raised when the user interrupts program execution, usually by pressing Ctrl+c.

Other Exceptions	Description
Exception	Base class for all exceptions. This catches most exception messages.
StopIteration	Raised when the next() method of an iterator does not point to any object.
AssertionError	Raised in case of failure of the Assert statement.
SystemExit	Raised when Python interpreter is quit by using the sys.exit() function. If not handled in the code, it causes the interpreter to exit.
OSError	Raises for operating system related errors.
EnvironmentError	Base class for all exceptions that occur outside the Python environment.
AttributeError	Raised in case of failure of an attribute reference or assignment.
NotImplementedError	Raised when an abstract method that needs to be implemented in an inherited class is not actually implemented.

All exceptions are objects. The classes that define the objects are organized in a hierarchy, which is shown below. This is important because the parent class of a set of related exceptions will catch all exception messages for itself and its child exceptions. For example, an `ArithmeticError` exception will catch itself and all `FloatingPointError`, `OverflowError`, and `ZeroDivisionError` exceptions.

```
BaseException
 +-- SystemExit
 +-- KeyboardInterrupt
 +-- GeneratorExit
 +-- Exception
      +-- StopIteration
      +-- StopAsyncIteration
      +-- ArithmeticError
```

```
|    +-- FloatingPointError
|    +-- OverflowError
|    +-- ZeroDivisionError
+-- AssertionError
+-- AttributeError
+-- BufferError
+-- EOFError
+-- ImportError
+-- LookupError
|    +-- IndexError
|    +-- KeyError
+-- MemoryError
+-- NameError
|    +-- UnboundLocalError
+-- OSError
|    +-- BlockingIOError
|    +-- ChildProcessError
|    +-- ConnectionError
|    |    +-- BrokenPipeError
|    |    +-- ConnectionAbortedError
|    |    +-- ConnectionRefusedError
|    |    +-- ConnectionResetError
|    +-- FileExistsError
|    +-- FileNotFoundError
|    +-- InterruptedError
|    +-- IsADirectoryError
|    +-- NotADirectoryError
|    +-- PermissionError
|    +-- ProcessLookupError
|    +-- TimeoutError
+-- ReferenceError
+-- RuntimeError
|    +-- NotImplementedError
|    +-- RecursionError
+-- SyntaxError
|    +-- IndentationError
|         +-- TabError
+-- SystemError
+-- TypeError
+-- ValueError
|    +-- UnicodeError
|         +-- UnicodeDecodeError
|         +-- UnicodeEncodeError
|         +-- UnicodeTranslateError
+-- Warning
     +-- DeprecationWarning
     +-- PendingDeprecationWarning
     +-- RuntimeWarning
     +-- SyntaxWarning
     +-- UserWarning
     +-- FutureWarning
     +-- ImportWarning
     +-- UnicodeWarning
     +-- BytesWarning
     +-- ResourceWarning
```

You have attempted 1 of 1 activities on this page

19.3. When to use try/except">

When to use try/except">

19.5. Exercises">

✔ Completed. Well Done!

19.5. Exercises">Next Section - 19.5. Exercises

| Back to top