Exceptions Reference

Exception	Meaning	Example cause
`FileNotFoundError`	A requested file does not exist	`open('missing.txt')`
`ValueError`	A value has the right type but unsuitable contents	`int('oops')`
`TypeError`	An operation cannot use the supplied type or arguments	`len(10)` or `len()`
`NameError`	A name has not been defined	`opne('scores.txt')`
`AttributeError`	An object has no requested attribute or method	`'10'.append('20')`
`IndexError`	A sequence index is out of range	`[][0]`
`KeyError`	A dictionary has no requested key	`{}['Sam']`
`ZeroDivisionError`	Division or remainder uses a zero divisor	`10 / 0`, `10 // 0`, or `10 % 0`
`AssertionError`	An assertion's condition is false	`assert 2 < 1`

Handling Exceptions

Statement	Behavior
`try:`	Begin the operations whose exceptions may be handled.
`except ValueError:`	Handle this exception type if it occurs in the try block.
`except (ValueError, TypeError):`	Use one handler for either exception type.
`except ValueError as error:`	Store the exception in error, including its message.
`raise ValueError(message)`	Stop the current operation by raising an exception.
`assert condition`	Raise AssertionError if the condition is false.
`pass`	Do nothing.
`continue`	Start the next iteration of the loop.
`break`	Leave the loop.