
GoCode
We learn by doing, by falling down, and by picking ourselves back up
HTTP://GOCODENOW.COM

Exceptions

1. What is it?
2. Three Different Examples
3. When to Use

What is it?

Definition: It is an event, which occurs during the execution of the program, that disrupts the normal flow.

try:
   You do your operations here;
   .....
except *ExceptionType, Argument*:
   You can print value of Argument here...

Exceptions

```
try:
    value = 5 / 1
except:
        print "Can't divide by
         zero"
else:
    print "Everything worked"
```

Exceptions

```
try:
    value = 5 / 1
finally:
    print "No matter what happens, this
        line will execute"
```

EXCEPTION NAME	DESCRIPTION
Exception	Base class for all exceptions
StopIteration	Raised when the next() method of an iterator does not point to any object.
SystemExit	Raised by the sys.exit() function.
StandardError	Base class for all built-in exceptions except StopIteration and SystemExit.
ArithmeticError	Base class for all errors that occur for numeric calculation.
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.
AssertionError	Raised in case of failure of the Assert statement.
AttributeError	Raised in case of failure of attribute reference or assignment.
EOFError	Raised when there is no input from either the raw_input() or input() function and the end of file is reache
ImportError	Raised when an import statement fails.
KeyboardInterrupt	Raised when the user interrupts program execution, usually by pressing Ctrl+c.
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.
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.
EnvironmentError	Base class for all exceptions that occur outside the Python environment.
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.
OSError	Raised for operating system-related errors.
SyntaxError	Raised when there is an error in Python syntax.
IndentationError	Raised when indentation is not specified properly.
SystemError	Raised when the interpreter finds an internal problem, but when this error is encountered the Python interpreter does not exit.
SystemExit	Raised when Python interpreter is quit by using the sys.exit() function. If not handled in the code, cause the interpreter to exit.

Exceptions

```
try:
    x = float(raw_input("Your number: "))
    inverse = 1.0 / x
except ValueError:
    print "You should have given either an int
        or a float"
except ZeroDivisionError:
    print "Infinity"
finally:
    print("There may or may not have been an
        exception.")
```

Use Sparingly

Capstone Project:

1) User Input (Integer/String)

2) Does File Exist?

Key Points

1) Exceptions are used to recover from errors.

2) Use sparingly – Only when you want program to continue and avoid a crash