University of Zurich[UZH]

Grid Computing Competence Center

Exceptions

GC3: Grid Computing Competence Center,
University of Zurich

Sep. 28, 2012

Exceptions

Exceptions are objects that inherit from the built-in `Exception` class.

To create a new exception just make a new class:

```
class NewKindOfError(Exception):
    """
    Do use the docstring to document
    what this error is about.
    """
    pass
```

Exceptions are handled by class name, so they usually do not need any new methods (although you are free to define some if needed).

See also: http://docs.python.org/library/exceptions.html

```
try:
  # code that might raise an exception
except SomeException:
  # handle some exception
except AnotherException, ex:
  # the actual Exception instance
  # is available as variable 'ex'
else:
  # performed on normal exit from 'try'
finally:
  # performed on exit in any case
```

The optional **`else`** clause is executed if and when control flows off the *end* of the **`try`** clause.

The optional **`finally`** clause is executed on exit from the **`try`** or **`except`** block in *any* case.

Reference: http://docs.python.org/reference/compound_stmts.html#try

Raising exceptions

Use the **raise** statement with an `Exception` instance:

```
if an_error_occurred:
  raise AnError("Spider sense is tingling.")
```

Within an **except** clause, you can use **raise** with no arguments to re-raise the current exception:

```
try:
  something()
except ItDidntWork:
  do_cleanup()
  # re-raise exception to caller
  raise
```

Exercise A: The `os.mkdir()` function raises a `OSError` exception if asked to create a directory that already exists.

Write a `mkdir_p(path)` function that creates a directory at `path`, but does nothing if the directory already exists. Return `True` if the directory has been actually created, and `False` if nothing was changed on the file system.

Exception handling example

Read lines from a CSV file, ignoring those that do not have the required number of fields. If other errors occur, abort. Close the file when done.

```
job_state = { } # empty dict
try:
  csv_file = open('jobs.csv', 'r')
  for line in csv_file:
    line = line.strip() # remove trailing newline
    try:
      name, jobid, state = line.split(",")
    except ValueError:
      continue # ignore line
    job_state[jobid] = state
except IOError:
  raise # up to caller
finally:
  csv_file.close()
```