University of Zurich[UZH]

Grid Computing Competence Center

String manipulation and file I/O

GC3: Grid Computing Competence Center,
University of Zurich

Sep. 28, 2012

File I/O, I

`stream = open(path,mode)`
Return a Python `file` object for reading or writing the file located at `path`. Mode is one of '`r`', '`w`' or '`a`' for reading, writing (truncates on open), appending. You can add a '+' character to enable read+write (other effects being the same).

***`stream`*`.close()`**
Close an open file.

`for line in stream:`
Loop over lines in the file one by one.

Reference:
http://docs.python.org/library/stdtypes.html#file-objects

File I/O, II

The `read(n)` method can be used to read *at most* `n` bytes from a file-like object:

```
>>> s = stream.read(2)
>>> s == 'py'
True
```

If `n` is omitted, `read()` reads until end-of-file.

Reference:
http://docs.python.org/library/stdtypes.html#file-objects

Exercise A: Write a function `load_data(filename)` that reads a file containing one integer number per line, and return a list of the integer values.

Test it with the values.dat file:

```
>>> load_data('values.dat')
[299850, 299740, 299900, 300070, 299930]
```

Operations on strings, I

`s.capitalize()`, `s.lower()`, `s.upper()`
Return a *copy* of the string capitalized / turned all lowercase / turned all uppercase.

`s.split(t)`
Split `s` at every occurrence of `t` and return a list of parts. If `t` is omitted, split on whitespace.

`s.startswith(t)`, `s.endswith(t)`
Return `True` if `t` is the initial/final substring of `s`.

Reference:
http://docs.python.org/library/stdtypes.html#string-methods

Operations on strings, II

`S.replace(old, new)`
Return a *copy* of string `S` with all occurrences of substring `old` replaced by `new`.

`s.lstrip(), s.rstrip(), s.strip()`
Return a *copy* of the string with the leading / trailing / leading *and* trailing whitespace removed.

Reference:
http://docs.python.org/library/stdtypes.html#string-methods

Exercise B: Write a program that reads the euro.csv file, and populates a dictionary from it: currency names (first column) are the dictionary keys, conversion rates (second column) are the dictionary values.

Exercise C: Building upon the previous exercise, create a `rates[][]` 2D array that stores the convertion rate of two currencies given the name, e.g., `rate['ITL']['DEM']` gives the conversion rate of Italian Liras to Deutsche Marks.

Filesystem operations, I

These functions are available from the `os` module.

`os.getcwd()`, `os.chdir(path)`
Return the path to the current working directory / Change the current working directory to `path`.

`os.listdir(dir)`
Return list of entries in directory `dir` (omitting '.' and '..')

`os.mkdir(path)`
Create a directory; fails if the directory already exists. Assumes that all parent directories exist already.

`os.makedirs(path)`
Create a directory; no-op if the directory already exists. Creates all the intermediate-level directories needed to contain the leaf.

Reference: http://docs.python.org/library/os.html

Filesystem operations, II

These functions are available from the `os.path` module.

`os.path.exists(path)`, `os.path.isdir(path)`
Return `True` if `path` exists / is a directory / is a regular file.

`os.path.basename(path)`, `os.path.dirname(path)`
Return the base name (the part after the last '/' character) or the directory name (the part before the last / character).

`os.path.abspath(path)`
Make `path` absolute (i.e., start with a /).

Reference: http://docs.python.org/library/os.path.html

Exercise D: *(Homework)* Write a Python program `rename.py` with the following command-line:

```
python rename.py EXT1 EXT2 DIR [DIR ...]
```

where:

ext1,ext2 Are file name extensions (without the leading dot), e.g., `jpg` and `jpeg`.

dir Is directory path; possibly, many directories names can be given on the command-line.

The `rename.py` command should rename all files in directory DIR, that end with extension `ext1` to end with extension `ext2` instead.