linecache — Read Text Files Efficiently

Purpose: Retrieve lines of text from files or imported Python modules, holding a cache of the results to make reading many lines from the same file more efficient.

The `linecache` module is used within other parts of the Python standard library when dealing with Python source files. The implementation of the cache holds the contents of files, parsed into separate lines, in memory. The API returns the requested line(s) by indexing into a `list`, and saves time over repeatedly reading the file and parsing lines to find the one desired. This is especially useful when looking for multiple lines from the same file, such as when producing a traceback for an error report.

Test Data

This text produced by a Lorem Ipsum generator is used as sample input.

```
# linecache_data.py

import os
import tempfile

lorem = '''Lorem ipsum dolor sit amet, consectetuer
adipiscing elit.  Vivamus eget elit. In posuere mi non
risus. Mauris id quam posuere lectus sollicitudin
varius. Praesent at mi. Nunc eu velit. Sed augue massa,
fermentum id, nonummy a, nonummy sit amet, ligula. Curabitur
eros pede, egestas at, ultricies ac, apellentesque eu,
tellus.

Sed sed odio sed mi luctus mollis. Integer et nulla ac augue
convallis accumsan. Ut felis. Donec lectus sapien, elementum
nec, condimentum ac, interdum non, tellus. Aenean viverra,
mauris vehicula semper porttitor, ipsum odio consectetuer
lorem, ac imperdiet eros odio a sapien. Nulla mauris tellus,
aliquam non, egestas a, nonummy et, erat. Vivamus sagittis
porttitor eros.'''


def make_tempfile():
    fd, temp_file_name = tempfile.mkstemp()
    os.close(fd)
    with open(temp_file_name, 'wt') as f:
        f.write(lorem)
    return temp_file_name


def cleanup(filename):
    os.unlink(filename)
```

Reading Specific Lines

The line numbers of files read by the `linecache` module start with 1, but normally lists start indexing the array from 0.

```
# linecache_getline.py

import linecache
from linecache_data import *

filename = make_tempfile()

# Pick out the same line from source and cache.
# (Notice that linecache counts from 1)
print('SOURCE:')
print('{!r}'.format(lorem.split('\n')[4]))
```

```
print()
print('CACHE:')
print('{!r}'.format(linecache.getline(filename, 5)))

cleanup(filename)
```

Each line returned includes a trailing newline.

```
$ python3 linecache_getline.py

SOURCE:
'fermentum id, nonummy a, nonummy sit amet, ligula. Curabitur'

CACHE:
'fermentum id, nonummy a, nonummy sit amet, ligula. Curabitur\n'
```

Handling Blank Lines

The return value always includes the newline at the end of the line, so if the line is empty the return value is just the newline.

```
# linecache_empty_line.py

import linecache
from linecache_data import *

filename = make_tempfile()

# Blank lines include the newline
print('BLANK : {!r}'.format(linecache.getline(filename, 8)))

cleanup(filename)
```

Line eight of the input file contains no text.

```
$ python3 linecache_empty_line.py

BLANK : '\n'
```

Error Handling

If the requested line number falls out of the range of valid lines in the file, `getline()` returns an empty string.

```
# linecache_out_of_range.py

import linecache
from linecache_data import *

filename = make_tempfile()

# The cache always returns a string, and uses
# an empty string to indicate a line which does
# not exist.
not_there = linecache.getline(filename, 500)
print('NOT THERE: {!r} includes {} characters'.format(
    not_there, len(not_there)))

cleanup(filename)
```

The input file only has 15 lines, so requesting line 500 is like trying to read past the end of the file.

```
$ python3 linecache_out_of_range.py

NOT THERE: '' includes 0 characters
```

Reading from a file that does not exist is handled in the same way.

```
# linecache_missing_file.py
```

```
# linecache_missing_file.py

import linecache

# Errors are even hidden if linecache cannot find the file
no_such_file = linecache.getline(
    'this_file_does_not_exist.txt', 1,
)
print('NO FILE: {!r}'.format(no_such_file))
```

The module never raises an exception when the caller tries to read data.

```
$ python3 linecache_missing_file.py

NO FILE: ''
```

Reading Python Source Files

Since `linecache` is used so heavily when producing tracebacks, one of its key features is the ability to find Python source modules in the import path by specifying the base name of the module.

```
# linecache_path_search.py

import linecache
import os

# Look for the linecache module, using
# the built in sys.path search.
module_line = linecache.getline('linecache.py', 3)
print('MODULE:')
print(repr(module_line))

# Look at the linecache module source directly.
file_src = linecache.__file__
if file_src.endswith('.pyc'):
    file_src = file_src[:-1]
print('\nFILE:')
with open(file_src, 'r') as f:
    file_line = f.readlines()[2]
print(repr(file_line))
```

The cache population code in `linecache` searches `sys.path` for the named module if it cannot find a file with that name in the current directory. This example looks for `linecache.py`. Since there is no copy in the current directory, the file from the standard library is found instead.

```
$ python3 linecache_path_search.py

MODULE:
'This is intended to read lines from modules imported -- hence
if a filename\n'

FILE:
'This is intended to read lines from modules imported -- hence
if a filename\n'
```

See also

- Standard library documentation for linecache

fnmatch — Unix-style Glob Pattern Matching

tempfile — Temporary File System Objects

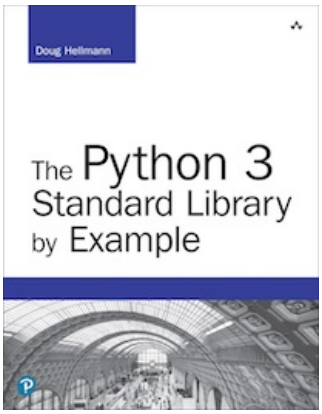
Quick Links

Test Data
Reading Specific Lines
Handling Blank Lines
Error Handling
Reading Python Source Files

This page was last updated 2016-12-18.

Navigation

fnmatch — Unix-style Glob Pattern Matching
tempfile — Temporary File System Objects



Get the book

The output from all the example programs from PyMOTW-3 has been generated with Python 3.7.1, unless otherwise noted. Some of the features described here may not be available in earlier versions of Python.

Looking for examples for Python 2?

This Site

Module Index
Index

Other Writing

Blog
The Python Standard Library By Example