Module Documentation For pytidylib

Release

Jason Stitt

April 07, 2009

Contents

The pytidylib module is a Python interface to the TidyLib library, which allows you to turn semi-valid HTML or XHTML code into valid code with an HTML 4, XHTML Transitional or XHTML Strict doctype. Some of the library's capabilities include:

- Clean up unclosed tags and unescaped characters such as ampersands
- Output HTML 4 or XHTML Transitional or Strict
- Convert named HTML entities to numeric entities, which are more portable and can be used in XML documents without an HTML doctype
- Clean up output from desktop programs such as Word
- Indent the output, including proper (i.e. no) indenting for `pre` elements

The pytidylib package is intended as a replacement for the uTidyLib package. The author found that uTidyLib has not been maintained in a while, and there are several outstanding patches. Compared to uTidyLib, the pytidylib package:

- Supports unicode strings
- Supports 64-bit systems and OS X
- Has improved performance, due to using cStringIO in place of StringIO, having the (optional) ability to re-use document objects, and a few other enhancements
- Does not leak memory when used repeatedly, due to proper freeing of document and error-reporting objects

This package relies on `ctypes`, which was added to Python in version 2.5. For versions 2.3 to 2.4, download `ctypes` from the ctypes home page.

1 Installing TidyLib

You must have TidyLib installed to use this Python module. There is no affiliation between the two projects; this is only a quick reference. How best to install TidyLib depends on your platform:

- Linux/BSD: First, try installing `tidylib` (or possibly `libtidy`) through your system's package management or ports system.
- OS X: You may already have TidyLib, especially if you have Apple's Developer Tools installed. In Terminal, run `locate libtidy` to find out.
- Windows: First, try installing the Windows package from the TidyLib homepage. As of this writing, the latest DLL version may not be fully up-to-date.
- If none of the above options works, downlaod the source code and build it yourself using the appropriate compiler for your platform. You must download the source using CVS:

  ```
  cvs -z3 -d:pserver:anonymous@tidy.cvs.sourceforge.net:/cvsroot/tidy co -P tidy
  ```

2 Installing pytidylib

Use easy_install:

```
easy_install pytidylib
```

Or, download the latest pytidylib from SourceForge and install in the usual way:

```
python setup.py install
```

3 Trivial example of use

The following code cleans up an invalid HTML document and sets an option:

```
from tidylib import tidy_document
document, errors = tidy_document('''<p>f&otilde;o <img src="bar.jpg">''',
    options={'numeric-entities':1})
print document
print errors
```

4 Configuration options

The Python interface allows you to pass options directly to libtidy. For a complete list of options, see the HTML Tidy Configuration Options Quick Reference or, from the command line, run `tidy -help-config`.

This module sets certain default options, as follows:

```
BASE_OPTIONS = {
    "output-xhtml": 1,      # XHTML instead of HTML4
    "indent": 1,            # Pretty; not too much of a performance hit
    "tidy-mark": 0,         # No tidy meta tag in output
    "wrap": 0,              # No wrapping
    "alt-text": "",         # Help ensure validation
    "doctype": 'strict',    # Little sense in transitional for tool-generated markup...
    "force-output": 1,      # May not get what you expect but you will get something
    }
```

If you do not like these options to be set for you, do the following after importing `tidylib`:

```
tidylib.BASE_OPTIONS = {}
```

5 Function reference

tidy_document(*text, options=None, keep_doc=False*)

Run a string with markup through Tidy and return the entire document.

text (str): The markup, which may be anything from an empty string to a complete XHTML document. Unicode values are supported; they will be encoded as utf-8, and tidylib's output will be decoded back to a unicode object.

options (dict): Options passed directly to tidylib; see the tidylib docs or run tidy -help-config from the command line.

keep_doc (boolean): If True, store 1 document object per thread and re-use it, for a slight performance boost especially when tidying very large numbers of very short documents.

-> (str, str): The tidied markup [0] and warning/error messages[1]. Warnings and errors are returned just as tidylib returns them.

tidy_fragment(*text, options=None, keep_doc=False*)

Tidy a string with markup and return it without the rest of the document. Tidy normally returns a full XHTML document; this function returns only the contents of the <body> element and is meant to be used for snippets. Calling tidy_fragment on elements that don't go in the <body>, like <title>, will produce odd behavior.

Arguments and return value as tidy_document. Note that tidy will always complain about the lack of a doctype and <title> element in fragments, and these errors are not stripped out for you.

release_tidy_doc()

Release the stored document object in the current thread. Only useful if you have called tidy_document or tidy_fragament with keep_doc=True.

Index