PyTidyLib documentation

Release

Jason Stitt

November 05, 2009

Contents

PyTidyLib is a Python package that wraps the HTML Tidy library. This allows you, from Python code, to "fix" invalid (X)HTML markup. Some of the library's many capabilities include:

- Clean up unclosed tags and unescaped characters such as ampersands
- Output HTML 4 or XHTML, strict or transitional, and add missing doctypes
- Convert named entities to numeric entities, which can then be used in XML documents without an HTML doctype.
- Clean up HTML from programs such as Word (to an extent)
- Indent the output, including proper (i.e. no) indenting for `pre` elements, which some (X)HTML indenting code overlooks.

PyTidyLib is intended as as replacement for uTidyLib, which fills a similar purpose. The author previously used uTidyLib but found several areas for improvement, including OS X support, 64-bit platform support, unicode support, fixing a memory leak, and better speed.

1 Naming conventions

HTML Tidy is a longstanding open-source library written in C that implements the actual functionality of cleaning up (X)HTML markup. It provides a shared library (`so`, `dll`, or `dylib`) that can variously be called `tidy`, `libtidy`,

or `tidylib`, as well as a command-line executable named `tidy`. For clarity, this document will consistently refer to it by the project name, HTML Tidy.

PyTidyLib is the name of the Python package discussed here. As this is the package name, `easy_install pytidylib` or `pip install pytidylib` is correct (they are case-insenstive). The *module* name is `tidylib`, so `import tidylib` is correct in Python code. This document will consistently use the package name, PyTidyLib, outside of code examples.

2 Installing HTML Tidy

You must have both HTML Tidy and PyTidyLib installed in order to use the functionality described here. There is no affiliation between the two projects. The following briefly outlines what you must do to install HTML Tidy. See the HTML Tidy web site for more information.

Linux/BSD or similar: First, try to use your distribution's package management system (`apt-get`, `yum`, etc.) to install HTML Tidy. It might go under the name `libtidy`, `tidylib`, `tidy`, or something similar. Otherwise see *Building from Source*, below.

OS X: You may already have HTML Tidy installed. In the Terminal, run `locate libtidy` and see if you get any results, which should end in `dylib`. Otherwise see *Building from Source*, below.

Windows: (Use PyTidyLib version 0.2 or later!) Prebuilt HTML Tidy DLLs are available from at least two locations. The int64.org Tidy Binaries page provides binaries that were built in 2005, for both 32-bit and 64-bit Windows, against a patched version of the source. The HTML Tidy web site links to a DLL built in 2006, for 32-bit Windows only, using the vanilla source (scroll near the bottom to "Other Builds" – use the one that reads "exe/lib/dll", *not* the "exe"-only version.)

Once you have a DLL (which may be named `tidy.dll`, `libtidy.dll`, or `tidylib.dll`), you must place it in a directory on your system path. If you are running Python from the command-line, placing the DLL in the present working directory will work, but this is unreliable otherwise (e.g. for server software).

See the articles How to set the path in Windows 2000/Windows XP (ComputerHope.com) and Modify a Users Path in Windows Vista (Question Defense) for more information on your system path.

Building from Source: The HTML Tidy developers have chosen to make the source code downloadable *only* through CVS, and not from the web site. Use the following CVS checkout at the command line:

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

Then see the instructions packaged with the source code or on the HTML Tidy web site.

3 Installing PyTidyLib

PyTidyLib is available on the Python Package Index and may be installed in the usual ways if you have pip or setuptools installed:

```
pip install pytidylib
# or:
easy_install pytidylib
```

You can also download the latest source distribution from the PyTidyLib web site.

4 Small 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
```

5 Configuration options

The Python interface allows you to pass options directly to HTML Tidy. 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 = {}
```

6 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