PyTidyLib: A Python Interface to HTML Tidy

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.

The maintenance of HTML Tidy has been transferred and is now active. Please see http://www.html-tidy.org/documentation/ for information about HTML Tidy.

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, `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.

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: You may be able to find prebuild DLLs. The DLL sources that were linked to in previous versions of this documentation have since gone 404 without obvious replacements.

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: Please see http://www.html-tidy.org/documentation/

Installing PyTidyLib

PyTidyLib is available on the Python Package Index:

```
pip install pytidylib
```

You can also download the latest source distribution from PyPI manually.

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 <
    options={'numeric-entities':1})
print document
print errors
```

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 = {
    "indent": 1,           # Pretty; not too much o
    "tidy-mark": 0,        # No tidy meta tag in ou
    "wrap": 0,             # No wrapping
    "alt-text": "",        # Help ensure validation
    "doctype": 'strict',   # Little sense in transi
    "force-output": 1,     # May not get what you e
    }
```

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

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

Function reference

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

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

tidylib.**release_tidy_doc**()