Webalin Documentation

Release 0.1.0-dev

Derek Payton

May 01, 2017

Contents

Webalin is a **Web A**ccessibility **Lin**ter that helps ensure your website is Section 508 compliant. Webalin is written in Python and can be called from the command line or imported for use in your projects test suite.

Author Derek Payton

Version 0.1.0

License MIT

CHAPTER 1

A Quick Demo

```
$ webalin http://python.org
E: 94: <input:domains> is missing <label>
E: 95: <input:sitesearch> is missing <label>
E: 96: <input:sourceid> is missing <label>
E: 97: <input:q> is missing <label>
E: 200: <input> is missing [id]
```

CHAPTER 2

Contents

Installation

Dependencies

- lxml
- requests

Install

Use pip to get the latest development version:

```
pip install -e git+git://github.com/dmpayton/webalin.git
```

Webalin will be uploaded to PyPI once it is deemed (relatively) stable.

Usage

Python Library

`webalin.analyze` can accept a URL or a full HTML document and returns errors as a list:

```
>>> import webalin
>>> webalin.analyze('https://python.org/')
{'messages': ['E: 94: <input:domains> is missing <label>',
              'E: 95: <input:sitesearch> is missing <label>',
              'E: 96: <input:sourceid> is missing <label>',
              'E: 97: <input:q> is missing <label>',
              'E: 200: <input> is missing [id]'],
 'stats': {'errors': 5, 'warnings': 0}}
```

```
>>> webalin.analyze(open('tests/resources/accessible.html', 'r').read())
{'messages': [], 'stats': {'errors': 0, 'warnings': 0}}
```

Command Line Utility

The command line utility accepts arguments from `sys.stdin` or `sys.argv` and prints errors to stdout:

```
$ cat tests/resources/inaccessible.html | ./webalin
E: -: <!DOCTYPE> is missing
E: -: <title> is missing
E: 7: <img:/satic/images/logo.png> is missing [alt]
E: 17: <table> is missing [summary]
E: 19: <th> is missing [scope]
E: 20: <th> is missing [scope]
E: 37: <input:id_username> is missing <label>
E: 41: <input:id_password> is missing <label>
W: 17: <table> contains no <tr> with [scope]
```

An HTML document that passes all accessibility tests will not return any output:

```
$ webalin https://www.djangoproject.com

$
```

Testing

```
cd tests
python tests.py
```

The unit tests are currently a very basic, only checking that an inaccessible document returns errors and an accessible one doesn't. Better unit tests would involve testing each accessibility test individually. This is still TODO.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search