# WebHandyTool

newAGEtech
Version 1
Sat Dec 5 2015

# Table of Contents

Table of contents

# Namespace Index

## Namespace List

Here is a list of all namespaces with brief descriptions:

# Hierarchical Index

## Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

object

TCPServer

TestCase

ThreadingMixIn

# Class Index

## Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

# File Index

## File List

Here is a list of all files with brief descriptions:

# Namespace Documentation

## pylinkvalidator Namespace Reference

### Namespaces

**POC Code**

**WebHandyTool**

## Variables

string **__version__** = "0.1"

## Variable Documentation

### string pylinkvalidator.__version__ = "0.1"

# pylinkvalidator.POC Code Namespace Reference

## Namespaces

**Web_Crawler**
**Web_Crawler_POC**

## Variables

string **__version__** = "0.1"

## Variable Documentation

### string pylinkvalidator.POC Code.__version__ = "0.1"

# pylinkvalidator.POC Code.Web_Crawler Namespace Reference

## Classes

class **HTML_corrector_help**
class **Web_Crawler**

## Variables

string **__author__** = 'Abe'
tuple **crawler** = **Web_Crawler**()
tuple **option** = crawler.option()

## Variable Documentation

### string pylinkvalidator.POC Code.Web_Crawler.__author__ = 'Abe'

### tuple pylinkvalidator.POC Code.Web_Crawler.crawler = Web_Crawler()

### tuple pylinkvalidator.POC Code.Web_Crawler.option = crawler.option()

# pylinkvalidator.POC Code.Web_Crawler_POC Namespace Reference

## Classes

class **Web_Crawler_POC**

## Variables

string **__author__** = 'Abe'
tuple **choice** = raw_input("Do you want to get all links on site (Enter : l) or download the site (Enter: d) ")
tuple **webcrawl** = **Web_Crawler_POC**()

## Variable Documentation

### string pylinkvalidator.POC Code.Web_Crawler_POC.__author__ = 'Abe'

### tuple pylinkvalidator.POC Code.Web_Crawler_POC.choice = raw_input("Do you want to get all links on site (Enter : l) or download the site (Enter: d) ")

### tuple pylinkvalidator.POC Code.Web_Crawler_POC.webcrawl = Web_Crawler_POC()

# pylinkvalidator.WebHandyTool Namespace Reference

## Namespaces

**config**
**download**
**errors**
**linkSearchAlgos**
**parsers**
**search**
**tests**
**urlCorrector**
**Web_Crawler**

## Variables

string **__version__** = "0.1"

## Variable Documentation

### string pylinkvalidator.WebHandyTool.__version__ = "0.1"

## pylinkvalidator.WebHandyTool.config Namespace Reference

### Functions

def **getConfig** ()

### Function Documentation

#### def pylinkvalidator.WebHandyTool.config.getConfig ()

## pylinkvalidator.WebHandyTool.download Namespace Reference

### Classes

class **download**

## pylinkvalidator.WebHandyTool.errors Namespace Reference

### Classes

class **errors**

## pylinkvalidator.WebHandyTool.linkSearchAlgos Namespace Reference

### Classes

class **linkSearchAlgos**

## pylinkvalidator.WebHandyTool.parsers Namespace Reference

### Classes

class **parsers**

## pylinkvalidator.WebHandyTool.search Namespace Reference

### Classes

class **search**

## pylinkvalidator.WebHandyTool.tests Namespace Reference

## Classes

class **crawlerTest**
class **searchTest**
class **ThreadedTCPServer**
class **urlCorrectorTest**

## Functions

def **start_http_server** ()

## Variables

tuple **TEST_FILES_DIR**
tuple **urlTestSuite** = unittest.TestLoader()
tuple **searchTestSuite** = unittest.TestLoader()
tuple **crawlTestSuite** = unittest.TestLoader()

## Detailed Description

```
Unit and integration tests for WebHandyTool
```

## Function Documentation

### def pylinkvalidator.WebHandyTool.tests.start_http_server ()

```
    Starts a simple http server for the test files
```

Here is the caller graph for this function:

IMAGE

## Variable Documentation

### tuple pylinkvalidator.WebHandyTool.tests.crawlTestSuite = unittest.TestLoader()

### tuple pylinkvalidator.WebHandyTool.tests.searchTestSuite = unittest.TestLoader()

### tuple pylinkvalidator.WebHandyTool.tests.TEST_FILES_DIR

**Initial value:**

```
= os.path.join(os.path.dirname(os.path.realpath(__file__)),
                               'testfiles')
```

### tuple pylinkvalidator.WebHandyTool.tests.urlTestSuite = unittest.TestLoader()

# pylinkvalidator.WebHandyTool.urlCorrector Namespace

## Reference

### Classes

class **HTML_corrector_help**
class **urlCorrector**

## pylinkvalidator.WebHandyTool.Web_Crawler Namespace Reference

### Classes

class **Web_Crawler**

### Variables

string **__author__** = 'Abe'
tuple **crawler** = **Web_Crawler**()
tuple **option** = crawler.option()

### Variable Documentation

**string pylinkvalidator.WebHandyTool.Web_Crawler.__author__ = 'Abe'**

**tuple pylinkvalidator.WebHandyTool.Web_Crawler.crawler = Web_Crawler()**

**tuple pylinkvalidator.WebHandyTool.Web_Crawler.option = crawler.option()**

## TestString Namespace Reference

### Classes

class **TestStringMethods**

# Class Documentation

## pylinkvalidator.WebHandyTool.tests.crawlerTest Class Reference

Inheritance diagram for pylinkvalidator.WebHandyTool.tests.crawlerTest:

IMAGE

Collaboration diagram for pylinkvalidator.WebHandyTool.tests.crawlerTest:

IMAGE

### Public Member Functions

def **setUpClass** (cls)
def **tearDownClass** (cls)
def **setUp** (self)
def **get_url** (self, test_url)
def **get_url_open** (self)
def **get_url_request** (self)
def **test_404_error** (self)
def **test_200_error** (self)
def **test_link_grabber** (self)
def **test_BFS_depth_0** (self)
def **test_BFS_depth_1** (self)
def **test_BFS_depth_2** (self)
def **test_BFS_depth_3** (self)
def **test_BFS_depth_4** (self)
def **test_BFS_depth_5** (self)
def **test_download** (self)

## Member Function Documentation

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.get_url ( *self*, *test_url*)

Here is the caller graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.get_url_open ( *self*)

Here is the caller graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.get_url_request ( *self*)

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.setUp ( *self*)

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.setUpClass ( *cls*)

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.tearDownClass ( *cls*)

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.test_200_error ( *self*)

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.test_404_error ( *self*)

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.test_BFS_depth_0 ( *self*)

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.test_BFS_depth_1 ( *self*)

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.test_BFS_depth_2 ( *self*)

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.test_BFS_depth_3 ( *self*)

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.test_BFS_depth_4 ( *self*)

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.test_BFS_depth_5 ( *self*)

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.test_download ( *self*)

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.tests.crawlerTest.test_link_grabber ( *self*)

Here is the call graph for this function:

IMAGE

**The documentation for this class was generated from the following file:**

WebHandyTool/**tests.py**

# pylinkvalidator.WebHandyTool.download.download Class Reference

Inheritance diagram for pylinkvalidator.WebHandyTool.download.download:

IMAGE

Collaboration diagram for pylinkvalidator.WebHandyTool.download.download:

IMAGE

## Public Member Functions

def **__init__** (self)<br>
def **download_resources** (self, **link**, file_type=None)

## Public Attributes

**config**<br>
**link**

## Constructor & Destructor Documentation

### def pylinkvalidator.WebHandyTool.download.download.__init__ ( *self*)

## Member Function Documentation

### def pylinkvalidator.WebHandyTool.download.download.download_resources ( *self*, *link*, *file_type* = `None`)

```
Writes all resources matching the given file type from the page link to the file
specified by destination.

:param link:
:param options:
:param file_type:
:return:
```

### Member Data Documentation

**pylinkvalidator.WebHandyTool.download.download.config**

**pylinkvalidator.WebHandyTool.download.download.link**

**The documentation for this class was generated from the following file:**

WebHandyTool/**download.py**

## pylinkvalidator.WebHandyTool.errors.errors Class Reference

Inheritance diagram for pylinkvalidator.WebHandyTool.errors.errors:

IMAGE

Collaboration diagram for pylinkvalidator.WebHandyTool.errors.errors:

IMAGE

### Public Member Functions

def **__init__** (self)
def **check_errors** (self, link, list_of_links=None)

### Public Attributes

**urlCorrect**

### Constructor & Destructor Documentation

**def pylinkvalidator.WebHandyTool.errors.errors.__init__ (** ***self*****)**

### Member Function Documentation

**def pylinkvalidator.WebHandyTool.errors.errors.check_errors (** ***self*****,** ***link*****,** ***list_of_links*** **= `None`)**

```
    Checks the all the links and reports the error message associated with
    all the links inputted

    :param link:
    :param list_of_links:
    :return List of Errors:
```

### Member Data Documentation

**pylinkvalidator.WebHandyTool.errors.errors.urlCorrect**

**The documentation for this class was generated from the following file:**

WebHandyTool/**errors.py**

## pylinkvalidator.POC Code.Web_Crawler.HTML_corrector_help Class Reference

Inheritance diagram for pylinkvalidator.POC Code.Web_Crawler.HTML_corrector_help:

IMAGE

Collaboration diagram for pylinkvalidator.POC Code.Web_Crawler.HTML_corrector_help:

IMAGE

### Public Member Functions

def **__init__** (self)
def **is_link** (self, url)
def **convert_iri_to_uri** (self, url_split)
def **url_encode_non_ascii** (self, url_part)

### Public Attributes

**NOT_LINK**

### Detailed Description

```
Library of helper functions that are used by HTML corrector to
fix url links.
```

### Constructor & Destructor Documentation

**def pylinkvalidator.POC Code.Web_Crawler.HTML_corrector_help.__init__ ( *self*)**

### Member Function Documentation

**def pylinkvalidator.POC Code.Web_Crawler.HTML_corrector_help.convert_iri_to_uri ( *self*, *url_split*)**

```
Attempts to convert potential IRI to URI.

IRI may contain non-ascii characters.
:param url_split:
:return:
```

Here is the call graph for this function:

IMAGE

#### def pylinkvalidator.POC Code.Web_Crawler.HTML_corrector_help.is_link ( *self*, *url*)

```
Return True if the url is not base 64 data or a local ref (#)

:param url:
:return Boolean either True or False:
```

#### def pylinkvalidator.POC Code.Web_Crawler.HTML_corrector_help.url_encode_non_ascii ( *self*, *url_part*)

```
For each byte in url_part, if the byte is outside ascii range, quote the
byte. UTF characters that take two bytes will be correctly converted using
this technique.

We do not quote the whole url part because it might contain already quoted
characters, which would then be double-quoted.

The url part is converted from utf-8 and then to utf-8, which might not
always work if there is mixed or bad encoding.
:param url_part:
:return:
```

Here is the caller graph for this function:

IMAGE

### Member Data Documentation

#### pylinkvalidator.POC Code.Web_Crawler.HTML_corrector_help.NOT_LINK

#### The documentation for this class was generated from the following file:

POC Code/**Web_Crawler.py**

## pylinkvalidator.WebHandyTool.urlCorrector.HTML_corrector_help Class Reference

Inheritance diagram for pylinkvalidator.WebHandyTool.urlCorrector.HTML_corrector_help:

IMAGE

Collaboration diagram for pylinkvalidator.WebHandyTool.urlCorrector.HTML_corrector_help:

IMAGE

## Public Member Functions

def **__init__** (self)
def **is_link** (self, url)
def **convert_iri_to_uri** (self, url_split)
def **url_encode_non_ascii** (self, url_part)

## Public Attributes

**NOT_LINK**

## Detailed Description

```
Library of helper functions that are used by HTML corrector to
fix url links.
```

## Constructor & Destructor Documentation

### def pylinkvalidator.WebHandyTool.urlCorrector.HTML_corrector_help.__init__ ( *self*)

## Member Function Documentation

### def pylinkvalidator.WebHandyTool.urlCorrector.HTML_corrector_help.convert_iri_to_uri ( *self*, *url_split*)

```
    Attempts to convert potential IRI to URI.

    IRI may contain non-ascii characters.
    :param url_split:
    :return:
```

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.urlCorrector.HTML_corrector_help.is_link ( *self*, *url*)

```
    Return True if the url is not base 64 data or a local ref (#)

    :param url:
    :return Boolean either True or False:
```

### def pylinkvalidator.WebHandyTool.urlCorrector.HTML_corrector_help.url_encode_non_ascii ( *self*, *url_part*)

```
    For each byte in url_part, if the byte is outside ascii range, quote the
    byte. UTF characters that take two bytes will be correctly converted using
    this technique.

    We do not quote the whole url part because it might contain already quoted
    characters, which would then be double-quoted.

    The url part is converted from utf-8 and then to utf-8, which might not
    always work if there is mixed or bad encoding.
    :param url_part:
    :return:
```

Here is the caller graph for this function:

IMAGE

## Member Data Documentation

### pylinkvalidator.WebHandyTool.urlCorrector.HTML_corrector_help.NOT_LINK

**The documentation for this class was generated from the following file:**

WebHandyTool/**urlCorrector.py**

# pylinkvalidator.WebHandyTool.linkSearchAlgos.linkSearchAlgos Class Reference

Inheritance diagram for pylinkvalidator.WebHandyTool.linkSearchAlgos.linkSearchAlgos:

IMAGE

Collaboration diagram for pylinkvalidator.WebHandyTool.linkSearchAlgos.linkSearchAlgos:

IMAGE

## Public Member Functions

def **__init__** (self)
def **depth_setter** (self, **depth**)
def **bfs** (self, link)
def **find_links** (self, link, destination=None)

## Public Attributes

**urlCorrect**
**HTML_help**

**parser**
**depth**

## Constructor & Destructor Documentation

### def pylinkvalidator.WebHandyTool.linkSearchAlgos.linkSearchAlgos.__init__ ( *self*)

## Member Function Documentation

### def pylinkvalidator.WebHandyTool.linkSearchAlgos.linkSearchAlgos.bfs ( *self*, *link*)

```
    Finds all the links on a give website using the BFS algorithm
    :param link:
    :return A list of all the links found by BFS:
```

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.linkSearchAlgos.linkSearchAlgos.depth_setter ( *self*, *depth*)

```
    Sets the default max depth variable for the web crawler
    :param depth:
    :return:
```

### def pylinkvalidator.WebHandyTool.linkSearchAlgos.linkSearchAlgos.find_links ( *self*, *link*, *destination* = `None`)

```
    Finds all the links (<a></a> anchor tags on page) on a page, also removes all
    the link that start with '#' or 'data:' as these are not valid urls

    :param link:
    :param destination:
    :return List of Links:
```

Here is the caller graph for this function:

IMAGE

## Member Data Documentation

### pylinkvalidator.WebHandyTool.linkSearchAlgos.linkSearchAlgos.depth

### pylinkvalidator.WebHandyTool.linkSearchAlgos.linkSearchAlgos.HTML_help

### pylinkvalidator.WebHandyTool.linkSearchAlgos.linkSearchAlgos.parser

**pylinkvalidator.WebHandyTool.linkSearchAlgos.linkSearchAlgos.urlCorrect**

**The documentation for this class was generated from the following file:**

WebHandyTool/**linkSearchAlgos.py**

## pylinkvalidator.WebHandyTool.parsers.parsers Class Reference

Inheritance diagram for pylinkvalidator.WebHandyTool.parsers.parsers:

IMAGE

Collaboration diagram for pylinkvalidator.WebHandyTool.parsers.parsers:

IMAGE

### Public Member Functions

def **BS_parse_data** (self, link)
def **HTML_text** (self, link)

### Member Function Documentation

**def pylinkvalidator.WebHandyTool.parsers.parsers.BS_parse_data (** ***self*****,** ***link*****)**

```
Returns BeautifulSoup object for the link given, this will allow modules parse through
pages data much faster

:param link:
:return BeautifulSoup :
```

**def pylinkvalidator.WebHandyTool.parsers.parsers.HTML_text (** ***self*****,** ***link*****)**

```
Returns HTML text data for Query search

:param link:
:return String :
```

**The documentation for this class was generated from the following file:**

WebHandyTool/**parsers.py**

## pylinkvalidator.WebHandyTool.search.search Class Reference

Inheritance diagram for pylinkvalidator.WebHandyTool.search.search:

IMAGE

Collaboration diagram for pylinkvalidator.WebHandyTool.search.search:

IMAGE

## Public Member Functions

def **__init__** (self)
def **query_search** (self, query, link, **list_of_links**=None, choice='exact')
def **exact_query** (self, query, data)
def **similar_query** (self, query, data, **proximity**)
def **whitespace_checker** (self, character)
def **setProximity** (self, **proximity**)

## Public Attributes

**list_of_links**
**SCHEME_HTTP**
**SCHEME_HTTPS**
**SUPPORTED_SCHEMES**
**parser**
**search_choices**
**search_choice**
**proximity**

## Constructor & Destructor Documentation

### def pylinkvalidator.WebHandyTool.search.search.__init__ ( *self*)

## Member Function Documentation

### def pylinkvalidator.WebHandyTool.search.search.exact_query ( *self*, *query*, *data*)

```
Searches through a String for a certain phrase or term. Returns the starting index for
all occurrences of the query String.
If the query is not located, it will return an empty array.

:param query - The String we are looking for:
:param data - The String we are searching through.:
:return Indexes corresponding to the beginning of the location of the String in
question.:
```

Here is the caller graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.search.search.query_search ( *self*, *query*, *link*, *list_of_links* = `None`, *choice* = `'exact'`)

```
Find queries

:param query:
:param data:
:param choice:
```

```
:return Query results:
```

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.search.search.setProximity ( *self*, *proximity*)

```
Set the search proximity
:param proximity:
:return:
```

### def pylinkvalidator.WebHandyTool.search.search.similar_query ( *self*, *query*, *data*, *proximity*)

```
Searches through a String for a certain phrase or term. Returns results that are close
to the query as well.
(i.e. "ap ple" or "bpple" would be noted for "apple") Returns the starting index for
all occurrences of Strings sufficiently close to
the query. If the query is not located, it will return an empty array.

:param: query  - The String we are looking for.
:param: data - The String we are searching through.:
:param: proximity - The size of the acceptable variation from the query.:
:return:Indexes corresponding to the beginning of the location of the String in
question.
```

Here is the call graph for this function:

IMAGE

Here is the caller graph for this function:

IMAGE

### def pylinkvalidator.WebHandyTool.search.search.whitespace_checker ( *self*, *character*)

```
Returns true if the character passed in is a whitespace character such as tab, space
or newline.

:param character - The character to be checked.:
:return boolean, if there is whitespace True,Whether the character is whitespace:
```

Here is the caller graph for this function:

IMAGE

## Member Data Documentation

### pylinkvalidator.WebHandyTool.search.search.list_of_links

### pylinkvalidator.WebHandyTool.search.search.parser

**pylinkvalidator.WebHandyTool.search.search.proximity**

**pylinkvalidator.WebHandyTool.search.search.SCHEME_HTTP**

**pylinkvalidator.WebHandyTool.search.search.SCHEME_HTTPS**

**pylinkvalidator.WebHandyTool.search.search.search_choice**

**pylinkvalidator.WebHandyTool.search.search.search_choices**

**pylinkvalidator.WebHandyTool.search.search.SUPPORTED_SCHEMES**

**The documentation for this class was generated from the following file:**

WebHandyTool/**search.py**

## pylinkvalidator.WebHandyTool.tests.searchTest Class Reference

Inheritance diagram for pylinkvalidator.WebHandyTool.tests.searchTest:

IMAGE

Collaboration diagram for pylinkvalidator.WebHandyTool.tests.searchTest:

IMAGE

### Public Member Functions

def **test_exact_string_search_1** (self)
def **test_exact_string_search_2** (self)
def **test_exact_string_search_3** (self)
def **test_similar_string_search_1** (self)
def **test_similar_string_search_2** (self)
def **test_similar_string_search_3** (self)
def **test_similar_string_search_4** (self)
def **test_similar_string_search_5** (self)

### Member Function Documentation

**def pylinkvalidator.WebHandyTool.tests.searchTest.test_exact_string_search_1 (** ***self*****)**

**def pylinkvalidator.WebHandyTool.tests.searchTest.test_exact_string_search_2 (** ***self*****)**

**def pylinkvalidator.WebHandyTool.tests.searchTest.test_exact_string_search_3 (** ***self*****)**

**def pylinkvalidator.WebHandyTool.tests.searchTest.test_similar_string_search_1 (** ***self*****)**

**def pylinkvalidator.WebHandyTool.tests.searchTest.test_similar_string_search_2 (** ***self*****)**

**def pylinkvalidator.WebHandyTool.tests.searchTest.test_similar_string_search_3 ( *self*)**

**def pylinkvalidator.WebHandyTool.tests.searchTest.test_similar_string_search_4 ( *self*)**

**def pylinkvalidator.WebHandyTool.tests.searchTest.test_similar_string_search_5 ( *self*)**

**The documentation for this class was generated from the following file:**

WebHandyTool/**tests.py**

## TestString.TestStringMethods Class Reference

Inheritance diagram for TestString.TestStringMethods:

IMAGE

Collaboration diagram for TestString.TestStringMethods:

IMAGE

### Public Member Functions

def **test_upper** (self)
def **test_isupper** (self)
def **test_split** (self)

### Member Function Documentation

**def TestString.TestStringMethods.test_isupper ( *self*)**

**def TestString.TestStringMethods.test_split ( *self*)**

**def TestString.TestStringMethods.test_upper ( *self*)**

**The documentation for this class was generated from the following file:**

InLabSubmissions/**TestString.py**

## pylinkvalidator.WebHandyTool.tests.ThreadedTCPServer Class Reference

Inheritance diagram for pylinkvalidator.WebHandyTool.tests.ThreadedTCPServer:

IMAGE

Collaboration diagram for pylinkvalidator.WebHandyTool.tests.ThreadedTCPServer:

IMAGE

The documentation for this class was generated from the following file:

WebHandyTool/**tests.py**

## pylinkvalidator.WebHandyTool.urlCorrector.urlCorrector Class Reference

Inheritance diagram for pylinkvalidator.WebHandyTool.urlCorrector.urlCorrector:

IMAGE

Collaboration diagram for pylinkvalidator.WebHandyTool.urlCorrector.urlCorrector:

IMAGE

### Public Member Functions

def **__init__** (self)<br>
def **HTML_corrector** (self, link)<br>
def **absolute_HTML_corrector** (self, link, base_link_split)

### Public Attributes

**list_of_links**<br>
**NOT_LINK**<br>
**SCHEME_HTTP**<br>
**SCHEME_HTTPS**<br>
**SUPPORTED_SCHEMES**

### Constructor & Destructor Documentation

#### def pylinkvalidator.WebHandyTool.urlCorrector.urlCorrector.__init__ ( *self*)

### Member Function Documentation

#### def pylinkvalidator.WebHandyTool.urlCorrector.urlCorrector.absolute_HTML_corrector ( *self*, *link*, *base_link_split*)

```
Takes in the base url and appends any relative or absolute links to the base urk.

:param link:
:param base_link_split:
:return Url object of split url result corrected link Ex; SplitResult(scheme=u'http',
netloc=u'canvasgroup.ca', path=u'/zdfzd', query=u'', fragment=u'') ::
```

Here is the call graph for this function:

IMAGE

#### def pylinkvalidator.WebHandyTool.urlCorrector.urlCorrector.HTML_corrector ( *self*, *link*)

```
Fixes the link passed in such that it becomes either a functioning link or is flagged
as a broken link.
:param link:
:return  Url object of split url result corrected link Ex; SplitResult(scheme=u'http',
netloc=u'canvasgroup.ca', path=u'/zdfzd', query=u'', fragment=u'') :
```

Here is the caller graph for this function:

IMAGE

### Member Data Documentation

**pylinkvalidator.WebHandyTool.urlCorrector.urlCorrector.list_of_links**

**pylinkvalidator.WebHandyTool.urlCorrector.urlCorrector.NOT_LINK**

**pylinkvalidator.WebHandyTool.urlCorrector.urlCorrector.SCHEME_HTTP**

**pylinkvalidator.WebHandyTool.urlCorrector.urlCorrector.SCHEME_HTTPS**

**pylinkvalidator.WebHandyTool.urlCorrector.urlCorrector.SUPPORTED_SCHEMES**

**The documentation for this class was generated from the following file:**

WebHandyTool/**urlCorrector.py**

## pylinkvalidator.WebHandyTool.tests.urlCorrectorTest Class Reference

Inheritance diagram for pylinkvalidator.WebHandyTool.tests.urlCorrectorTest:

IMAGE

Collaboration diagram for pylinkvalidator.WebHandyTool.tests.urlCorrectorTest:

IMAGE

### Public Member Functions

def **test_clean_url_split** (self)
def **test_get_absolute_url** (self)

### Member Function Documentation

**def pylinkvalidator.WebHandyTool.tests.urlCorrectorTest.test_clean_url_split (** ***self*****)**

**def pylinkvalidator.WebHandyTool.tests.urlCorrectorTest.test_get_absolute_url (** ***self*****)**

**The documentation for this class was generated from the following file:**

WebHandyTool/**tests.py**

## pylinkvalidator.POC Code.Web_Crawler.Web_Crawler Class Reference

Inheritance diagram for pylinkvalidator.POC Code.Web_Crawler.Web_Crawler:

IMAGE

Collaboration diagram for pylinkvalidator.POC Code.Web_Crawler.Web_Crawler:

IMAGE

### Public Member Functions

def **__init__** (self)
def **option** (self)
def **dfs** (self, **choice**)
def **bfs** (self, link)
def **HTML_corrector** (self, link)
def **absolute_HTML_corrector** (self, link, base_link_split)
def **download_resources** (self, link, options='-rA', file_type=None)
def **exact_query** (self, query, data)
def **similar_query** (self, query, data, proximity)
def **whitespace_checker** (self, character)
def **find_links** (self, link, destination=None)
def **check_errors** (self, link, **list_of_links**=None)
def **BS_parse_data** (self, link)
def **HTML_text** (self, link)
def **query_search** (self, query, link, **list_of_links**=None, **choice**='exact')
def **depth_setter** (self, **depth**)
def **website_Depth** (self, link, **depth**)

### Public Attributes

**depth**
**algo**
**choices**
**search_choices**
**choice**
**search_choice**
**output**
**seed**
**list_of_links**
**NOT_LINK**
**SCHEME_HTTP**
**SCHEME_HTTPS**
**SUPPORTED_SCHEMES**

### Detailed Description

```
Main class, it is for crawling.
Class variable includes depth,algo,choices,choice,output,seed,list of links
```

## Constructor & Destructor Documentation

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.__init__ ( *self*)

## Member Function Documentation

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.absolute_HTML_corrector ( *self*, *link*, *base_link_split*)

```
Takes in the base url and appends any relative or absolute links to the base urk.

:param link:
:param base_link_split:
:return Url object of split url result corrected link Ex; SplitResult(scheme=u'http',
netloc=u'canvasgroup.ca', path=u'/zdfzd', query=u'', fragment=u'') ::
```

Here is the call graph for this function:

IMAGE

Here is the caller graph for this function:

IMAGE

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.bfs ( *self*, *link*)

```
Finds all the links on a give website using the BFS algorithm
:param link:
:return A list of all the links found by BFS:
```

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.BS_parse_data ( *self*, *link*)

```
Returns BeautifulSoup object for the link given, this will allow modules parse through
pages data much faster

:param link:
:return BeautifulSoup :
```

Here is the caller graph for this function:

IMAGE

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.check_errors ( *self*, *link*, *list_of_links* = `None`)

```
Checks the all the links and reports the error message associated with
all the links inputted
```

```
    :param link:
    :param list_of_links:
    :return List of Errors:
```

Here is the call graph for this function:

IMAGE

Here is the caller graph for this function:

IMAGE

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.depth_setter ( *self*, *depth*)

```
    Sets the default max depth variable for the web crawler
    :param depth:
    :return:
```

Here is the caller graph for this function:

IMAGE

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.dfs ( *self*, *choice*)

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.download_resources ( *self*, *link*, *options* = `'-rA'`, *file_type* = `None`)

```
    Writes all resources matching the given file type from the page link to the file
    specified by destination.

    :param link:
    :param options:
    :param file_type:
    :return:
```

Here is the caller graph for this function:

IMAGE

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.exact_query ( *self*, *query*, *data*)

```
    Searches through a String for a certain phrase or term. Returns the starting index for
    all occurrences of the query String.
    If the query is not located, it will return an empty array.

    :param query - The String we are looking for:
    :param data - The String we are searching through.:
    :return Indexes corresponding to the beginning of the location of the String in
    question.:
```

Here is the caller graph for this function:

IMAGE

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.find_links ( *self*, *link*, *destination* = `None`)

```
Finds all the links (<a></a> anchor tags on page) on a page, also removes all
the link that start with '#' or 'data:' as these are not valid urls

:param link:
:param destination:
:return List of Links:
```

Here is the call graph for this function:

IMAGE

Here is the caller graph for this function:

IMAGE

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.HTML_corrector ( *self*, *link*)

```
Fixes the link passed in such that it becomes either a functioning link or is flagged
as a broken link.
:param link:
:return  Url object of split url result corrected link Ex; SplitResult(scheme=u'http',
netloc=u'canvasgroup.ca', path=u'/zdfzd', query=u'', fragment=u'') :
```

Here is the caller graph for this function:

IMAGE

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.HTML_text ( *self*, *link*)

```
Returns HTML text data for Query search

:param link:
:return String :
```

Here is the caller graph for this function:

IMAGE

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.option ( *self*)

```
Gets user input to set various program options related to
how the user would like to handle crawling a webpage.

Ask for users option choices

:return:
```

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.query_search ( *self*, *query*, *link*, *list_of_links* = `None`, *choice* = `'exact'`)

```
Find queries

:param query:
:param data:
:param choice:
:return Query results:
```

Here is the call graph for this function:

IMAGE

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.similar_query ( *self*, *query*, *data*, *proximity*)

```
Searches through a String for a certain phrase or term. Returns results that are close
to the query as well.
(i.e. "ap ple" or "bpple" would be noted for "apple") Returns the starting index for
all occurrences of Strings sufficiently close to
the query. If the query is not located, it will return an empty array.

:param: query  - The String we are looking for.
:param: data - The String we are searching through.:
:param: proximity - The size of the acceptable variation from the query.:
:return:Indexes corresponding to the beginning of the location of the String in
question.
```

Here is the call graph for this function:

IMAGE

Here is the caller graph for this function:

IMAGE

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.website_Depth ( *self*, *link*, *depth*)

```
It provides a structured model of the website and other site the initial site is
connect to. It displays
a hierarchy that will show users how crawled link interact with each other. Shows all
the depths

:param link:
:param depth:
:return A pretty print of Hierarchy:
```

Here is the caller graph for this function:

IMAGE

### def pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.whitespace_checker ( *self*, *character*)

```
Returns true if the character passed in is a whitespace character such as tab, space
or newline.

:param character - The character to be checked.:
:return boolean, if there is whitespace True,Whether the character is whitespace:
```

Here is the caller graph for this function:

IMAGE

### Member Data Documentation

#### pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.algo

#### pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.choice

#### pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.choices

#### pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.depth

#### pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.list_of_links

#### pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.NOT_LINK

#### pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.output

#### pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.SCHEME_HTTP

#### pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.SCHEME_HTTPS

#### pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.search_choice

#### pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.search_choices

#### pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.seed

#### pylinkvalidator.POC Code.Web_Crawler.Web_Crawler.SUPPORTED_SCHEMES

**The documentation for this class was generated from the following file:**

POC Code/**Web_Crawler.py**

## pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler Class Reference

Inheritance diagram for pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler:

IMAGE

Collaboration diagram for pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler:
IMAGE

## Public Member Functions

def **__init__** (self)
def **option** (self)

## Public Attributes

**urlCorrect**
**HTML_corrector_helper**
**download**
**errors**
**parsers**
**choices**
**choice**
**list_of_links**
**NOT_LINK**
**SCHEME_HTTP**
**SCHEME_HTTPS**
**SUPPORTED_SCHEMES**
**algo**

## Detailed Description

```
Main class, it is for crawling.
Class variable includes depth,algo,choices,choice,output,seed,list of links
```

## Constructor & Destructor Documentation

### def pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler.__init__ ( *self*)

## Member Function Documentation

### def pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler.option ( *self*)

```
Gets user input to set various program options related to
how the user would like to handle crawling a webpage.

Ask for users option choices

:return:
```

## Member Data Documentation

### pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler.algo

**pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler.choice**

**pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler.choices**

**pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler.download**

**pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler.errors**

**pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler.HTML_corrector_helper**

**pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler.list_of_links**

**pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler.NOT_LINK**

**pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler.parsers**

**pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler.SCHEME_HTTP**

**pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler.SCHEME_HTTPS**

**pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler.SUPPORTED_SCHEMES**

**pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler.urlCorrect**

**The documentation for this class was generated from the following file:**

WebHandyTool/**Web_Crawler.py**

# pylinkvalidator.POC Code.Web_Crawler_POC.Web_Crawler_POC Class Reference

Inheritance diagram for pylinkvalidator.POC Code.Web_Crawler_POC.Web_Crawler_POC:

IMAGE

Collaboration diagram for pylinkvalidator.POC Code.Web_Crawler_POC.Web_Crawler_POC:

IMAGE

## Public Member Functions

def **__init__** (self)
def **grabLinks** (self)
def **downloadHTML** (self)

## Public Attributes

**start**

### Constructor & Destructor Documentation

**def pylinkvalidator.POC Code.Web_Crawler_POC.Web_Crawler_POC.__init__ (** ***self*****)**

### Member Function Documentation

**def pylinkvalidator.POC Code.Web_Crawler_POC.Web_Crawler_POC.downloadHTML (** ***self*****)**

**def pylinkvalidator.POC Code.Web_Crawler_POC.Web_Crawler_POC.grabLinks (** ***self*****)**

### Member Data Documentation

**pylinkvalidator.POC Code.Web_Crawler_POC.Web_Crawler_POC.start**

**The documentation for this class was generated from the following file:**

- POC Code/**Web_Crawler_POC.py**

# File Documentation

## __init__.py File Reference

### Namespaces

**pylinkvalidator**

### Variables

string **pylinkvalidator.__version__** = "0.1"

## POC Code/__init__.py File Reference

### Namespaces

**pylinkvalidator.POC Code**

### Variables

string **pylinkvalidator.POC Code.__version__** = "0.1"

## WebHandyTool/__init__.py File Reference

### Namespaces

**pylinkvalidator.WebHandyTool**

### Variables

string **pylinkvalidator.WebHandyTool.__version__** = "0.1"

## InLabSubmissions/TestString.py File Reference

### Classes

class **TestString.TestStringMethods**

### Namespaces

**TestString**

## POC Code/Web_Crawler.py File Reference

### Classes

class **pylinkvalidator.POC Code.Web_Crawler.Web_Crawler**
class **pylinkvalidator.POC Code.Web_Crawler.HTML_corrector_help**

### Namespaces

**pylinkvalidator.POC Code.Web_Crawler**

### Variables

string **pylinkvalidator.POC Code.Web_Crawler.__author__** = 'Abe'
tuple **pylinkvalidator.POC Code.Web_Crawler.crawler** = Web_Crawler()
tuple **pylinkvalidator.POC Code.Web_Crawler.option** = crawler.option()

## WebHandyTool/Web_Crawler.py File Reference

### Classes

class **pylinkvalidator.WebHandyTool.Web_Crawler.Web_Crawler**

### Namespaces

**pylinkvalidator.WebHandyTool.Web_Crawler**

### Variables

string **pylinkvalidator.WebHandyTool.Web_Crawler.__author__** = 'Abe'
tuple **pylinkvalidator.WebHandyTool.Web_Crawler.crawler** = Web_Crawler()
tuple **pylinkvalidator.WebHandyTool.Web_Crawler.option** = crawler.option()

## POC Code/Web_Crawler_POC.py File Reference

### Classes

class **pylinkvalidator.POC Code.Web_Crawler_POC.Web_Crawler_POC**

### Namespaces

**pylinkvalidator.POC Code.Web_Crawler_POC**

### Variables

string **pylinkvalidator.POC Code.Web_Crawler_POC.__author__** = 'Abe'
tuple **pylinkvalidator.POC Code.Web_Crawler_POC.choice** = raw_input("Do you want to get all links on site (Enter : l) or download the site (Enter: d) ")
tuple **pylinkvalidator.POC Code.Web_Crawler_POC.webcrawl** = Web_Crawler_POC()

## WebHandyTool/config.py File Reference

### Namespaces

**pylinkvalidator.WebHandyTool.config**

### Functions

def **pylinkvalidator.WebHandyTool.config.getConfig** ()

## WebHandyTool/download.py File Reference

### Classes

class **pylinkvalidator.WebHandyTool.download.download**

### Namespaces

**pylinkvalidator.WebHandyTool.download**

## WebHandyTool/errors.py File Reference

### Classes

class **pylinkvalidator.WebHandyTool.errors.errors**

### Namespaces

**pylinkvalidator.WebHandyTool.errors**

## WebHandyTool/linkSearchAlgos.py File Reference

### Classes

class **pylinkvalidator.WebHandyTool.linkSearchAlgos.linkSearchAlgos**

### Namespaces

**pylinkvalidator.WebHandyTool.linkSearchAlgos**

# WebHandyTool/parsers.py File Reference

## Classes

class **pylinkvalidator.WebHandyTool.parsers.parsers**

## Namespaces

**pylinkvalidator.WebHandyTool.parsers**

# WebHandyTool/search.py File Reference

## Classes

class **pylinkvalidator.WebHandyTool.search.search**

## Namespaces

**pylinkvalidator.WebHandyTool.search**

# WebHandyTool/tests.py File Reference

## Classes

class **pylinkvalidator.WebHandyTool.tests.ThreadedTCPServer**
class **pylinkvalidator.WebHandyTool.tests.urlCorrectorTest**
class **pylinkvalidator.WebHandyTool.tests.searchTest**
class **pylinkvalidator.WebHandyTool.tests.crawlerTest**

## Namespaces

**pylinkvalidator.WebHandyTool.tests**

## Functions

def **pylinkvalidator.WebHandyTool.tests.start_http_server** ()

## Variables

tuple **pylinkvalidator.WebHandyTool.tests.TEST_FILES_DIR**
tuple **pylinkvalidator.WebHandyTool.tests.urlTestSuite** = unittest.TestLoader()
tuple **pylinkvalidator.WebHandyTool.tests.searchTestSuite** = unittest.TestLoader()
tuple **pylinkvalidator.WebHandyTool.tests.crawlTestSuite** = unittest.TestLoader()

# WebHandyTool/urlCorrector.py File Reference

## Classes

class **pylinkvalidator.WebHandyTool.urlCorrector.urlCorrector**
class **pylinkvalidator.WebHandyTool.urlCorrector.HTML_corrector_help**

## Namespaces

**pylinkvalidator.WebHandyTool.urlCorrector**

# Index

INDEX