# :mod:`shlex` --- Simple lexical analysis

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 1); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 4)

Unknown directive type "module".

```
.. module:: shlex
   :synopsis: Simple lexical analysis for Unix shell-like languages.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 7)

Unknown directive type "moduleauthor".

```
.. moduleauthor:: Eric S. Raymond <esr@snark.thyrsus.com>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 8)

Unknown directive type "moduleauthor".

```
.. moduleauthor:: Gustavo Niemeyer <niemeyer@conectiva.com>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 9)

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Eric S. Raymond <esr@snark.thyrsus.com>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 10)

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Gustavo Niemeyer <niemeyer@conectiva.com>
```

**Source code:** :source:`Lib/shlex.py`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 12); *backlink*

Unknown interpreted text role "source".

---

The :class:`~shlex.shlex` class makes it easy to write lexical analyzers for simple syntaxes resembling that of the Unix shell. This will often be useful for writing minilanguages, (for example, in run control files for Python applications) or for parsing quoted strings.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 16); *backlink*

Unknown interpreted text role "class".

The :mod:`shlex` module defines the following functions:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

**main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 21); *backlink***

Unknown interpreted text role "mod".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 24)**

Unknown directive type "function".

```
.. function:: split(s, comments=False, posix=True)

   Split the string *s* using shell-like syntax. If *comments* is :const:`False`
   (the default), the parsing of comments in the given string will be disabled
   (setting the :attr:`~shlex.commenters` attribute of the
   :class:`~shlex.shlex` instance to the empty string).  This function operates
   in POSIX mode by default, but uses non-POSIX mode if the *posix* argument is
   false.

   .. note::

      Since the :func:`split` function instantiates a :class:`~shlex.shlex`
      instance, passing ``None`` for *s* will read the string to split from
      standard input.

   .. deprecated:: 3.9
      Passing ``None`` for *s* will raise an exception in future Python
      versions.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 43)**

Unknown directive type "function".

```
.. function:: join(split_command)

   Concatenate the tokens of the list *split_command* and return a string.
   This function is the inverse of :func:`split`.

      >>> from shlex import join
      >>> print(join(['echo', '-n', 'Multiple words']))
      echo -n 'Multiple words'

   The returned value is shell-escaped to protect against injection
   vulnerabilities (see :func:`quote`).

   .. versionadded:: 3.8
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 58)**

Unknown directive type "function".

```
.. function:: quote(s)

   Return a shell-escaped version of the string *s*.  The returned value is a
   string that can safely be used as one token in a shell command line, for
   cases where you cannot use a list.

   .. _shlex-quote-warning:

   .. warning::

      The ``shlex`` module is **only designed for Unix shells**.

      The :func:`quote` function is not guaranteed to be correct on non-POSIX
      compliant shells or shells from other operating systems such as Windows.
      Executing commands quoted by this module on such shells can open up the
      possibility of a command injection vulnerability.

      Consider using functions that pass command arguments with lists such as
      :func:`subprocess.run` with ``shell=False``.

   This idiom would be unsafe:

      >>> filename = 'somefile; rm -rf ~'
```

```
   >>> command = 'ls -l {}'.format(filename)
   >>> print(command)  # executed by a shell: boom!
   ls -l somefile; rm -rf ~

:func:`quote` lets you plug the security hole:

   >>> from shlex import quote
   >>> command = 'ls -l {}'.format(quote(filename))
   >>> print(command)
   ls -l 'somefile; rm -rf ~'
   >>> remote_command = 'ssh home {}'.format(quote(command))
   >>> print(remote_command)
   ssh home 'ls -l '"'"'somefile; rm -rf ~'"'"''

The quoting is compatible with UNIX shells and with :func:`split`:

   >>> from shlex import split
   >>> remote_command = split(remote_command)
   >>> remote_command
   ['ssh', 'home', "ls -l 'somefile; rm -rf ~'"]
   >>> command = split(remote_command[-1])
   >>> command
   ['ls', '-l', 'somefile; rm -rf ~']

.. versionadded:: 3.3
```

The :mod:`shlex` module defines the following class:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 107); *backlink***

Unknown interpreted text role "mod".

A :class:`~shlex.shlex` instance or subclass instance is a lexical analyzer object. The initialization argument, if present, specifies where to read characters from. It must be a file-/stream-like object with :meth:`~io.TextIOBase.read` and :meth:`~io.TextIOBase.readline` methods, or a string. If no argument is given, input will be taken from `sys.stdin`. The second optional argument is a filename string, which sets the initial value of the :attr:`~shlex.infile` attribute. If the *instream* argument is omitted or equal to `sys.stdin`, this second argument defaults to "stdin". The *posix* argument defines the operational mode: when *posix* is not true (default), the :class:`~shlex.shlex` instance will operate in compatibility mode. When operating in POSIX mode, :class:`~shlex.shlex` will try to be as close as possible to the POSIX shell parsing rules. The *punctuation_chars* argument provides a way to make the behaviour even closer to how real shells parse. This can take a number of values: the default value, `False`, preserves the behaviour seen under Python 3.5 and earlier. If set to `True`, then parsing of the characters `();<>|&` is changed: any run of these characters (considered punctuation characters) is returned as a single token. If set to a non-empty string of characters, those characters will be used as the punctuation characters. Any characters in the :attr:`wordchars` attribute that appear in *punctuation_chars* will be removed from :attr:`wordchars`. See :ref:`improved-shell-compatibility` for more information. *punctuation_chars* can be set only upon :class:`~shlex.shlex` instance creation and can't be modified later.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 112); *backlink***

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 112); *backlink***

Unknown interpreted text role "meth".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 112); *backlink***

Unknown interpreted text role "meth".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 112); *backlink***

Unknown interpreted text role "attr".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 112); *backlink***

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 112); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 112); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 112); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 112); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 112); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 137)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.6
   The *punctuation_chars* parameter was added.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 140)

Unknown directive type "seealso".

```
.. seealso::

   Module :mod:`configparser`
      Parser for configuration files similar to the Windows :file:`.ini` files.
```

## shlex Objects

A :class:`~shlex.shlex` instance has the following methods:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 151); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 154)

Unknown directive type "method".

```
.. method:: shlex.get_token()

   Return a token.  If tokens have been stacked using :meth:`push_token`, pop a
   token off the stack.  Otherwise, read one from the input stream.  If reading
   encounters an immediate end-of-file, :attr:`eof` is returned (the empty
   string (``''``) in non-POSIX mode, and ``None`` in POSIX mode).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 162)

Unknown directive type "method".

```
.. method:: shlex.push_token(str)

   Push the argument onto the token stack.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 167)

Unknown directive type "method".

```
.. method:: shlex.read_token()

   Read a raw token.  Ignore the pushback stack, and do not interpret source
   requests.  (This is not ordinarily a useful entry point, and is documented here
   only for the sake of completeness.)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 174)

Unknown directive type "method".

```
.. method:: shlex.sourcehook(filename)

   When :class:`~shlex.shlex` detects a source request (see :attr:`source`
   below) this method is given the following token as argument, and expected
   to return a tuple consisting of a filename and an open file-like object.

   Normally, this method first strips any quotes off the argument.  If the result
   is an absolute pathname, or there was no previous source request in effect, or
   the previous source was a stream (such as ``sys.stdin``), the result is left
   alone.  Otherwise, if the result is a relative pathname, the directory part of
   the name of the file immediately before it on the source inclusion stack is
   prepended (this behavior is like the way the C preprocessor handles ``#include
   "file.h"``).

   The result of the manipulations is treated as a filename, and returned as the
   first component of the tuple, with :func:`open` called on it to yield the second
   component. (Note: this is the reverse of the order of arguments in instance
   initialization!)

   This hook is exposed so that you can use it to implement directory search paths,
   addition of file extensions, and other namespace hacks. There is no
   corresponding 'close' hook, but a shlex instance will call the
   :meth:`~io.IOBase.close` method of the sourced input stream when it returns
   EOF.

   For more explicit control of source stacking, use the :meth:`push_source` and
   :meth:`pop_source` methods.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 203)

Unknown directive type "method".

```
.. method:: shlex.push_source(newstream, newfile=None)

   Push an input source stream onto the input stack.  If the filename argument is
   specified it will later be available for use in error messages.  This is the
   same method used internally by the :meth:`sourcehook` method.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

**main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 210)**

Unknown directive type "method".

```
.. method:: shlex.pop_source()

   Pop the last-pushed input source from the input stack. This is the same method
   used internally when the lexer reaches EOF on a stacked input stream.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 216)**

Unknown directive type "method".

```
.. method:: shlex.error_leader(infile=None, lineno=None)

   This method generates an error message leader in the format of a Unix C compiler
   error label; the format is ``'"%s", line %d: '``, where the ``%s`` is replaced
   with the name of the current source file and the ``%d`` with the current input
   line number (the optional arguments can be used to override these).

   This convenience is provided to encourage :mod:`shlex` users to generate error
   messages in the standard, parseable format understood by Emacs and other Unix
   tools.
```

Instances of :class:`~shlex.shlex` subclasses have some public instance variables which either control lexical analysis or can be used for debugging:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 227); *backlink***

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 231)**

Unknown directive type "attribute".

```
.. attribute:: shlex.commenters

   The string of characters that are recognized as comment beginners. All
   characters from the comment beginner to end of line are ignored. Includes just
   ``'#'`` by default.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 238)**

Unknown directive type "attribute".

```
.. attribute:: shlex.wordchars

   The string of characters that will accumulate into multi-character tokens.  By
   default, includes all ASCII alphanumerics and underscore.  In POSIX mode, the
   accented characters in the Latin-1 set are also included.  If
   :attr:`punctuation_chars` is not empty, the characters ``~-./*?=``, which can
   appear in filename specifications and command line parameters, will also be
   included in this attribute, and any characters which appear in
   ``punctuation_chars`` will be removed from ``wordchars`` if they are present
   there. If :attr:`whitespace_split` is set to ``True``, this will have no
   effect.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 251)**

Unknown directive type "attribute".

```
.. attribute:: shlex.whitespace

   Characters that will be considered whitespace and skipped.  Whitespace bounds
```

```
tokens.  By default, includes space, tab, linefeed and carriage-return.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 257)**

Unknown directive type "attribute".

```
.. attribute:: shlex.escape

   Characters that will be considered as escape. This will be only used in POSIX
   mode, and includes just ``'\'`` by default.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 263)**

Unknown directive type "attribute".

```
.. attribute:: shlex.quotes

   Characters that will be considered string quotes.  The token accumulates until
   the same quote is encountered again (thus, different quote types protect each
   other as in the shell.)  By default, includes ASCII single and double quotes.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 270)**

Unknown directive type "attribute".

```
.. attribute:: shlex.escapedquotes

   Characters in :attr:`quotes` that will interpret escape characters defined in
   :attr:`escape`.  This is only used in POSIX mode, and includes just ``'"'`` by
   default.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 277)**

Unknown directive type "attribute".

```
.. attribute:: shlex.whitespace_split

   If ``True``, tokens will only be split in whitespaces.  This is useful, for
   example, for parsing command lines with :class:`~shlex.shlex`, getting
   tokens in a similar way to shell arguments.  When used in combination with
   :attr:`punctuation_chars`, tokens will be split on whitespace in addition to
   those characters.

   .. versionchanged:: 3.8
      The :attr:`punctuation_chars` attribute was made compatible with the
      :attr:`whitespace_split` attribute.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 290)**

Unknown directive type "attribute".

```
.. attribute:: shlex.infile

   The name of the current input file, as initially set at class instantiation time
   or stacked by later source requests.  It may be useful to examine this when
   constructing error messages.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-**

**main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 297)**

Unknown directive type "attribute".

```
.. attribute:: shlex.instream

   The input stream from which this :class:`~shlex.shlex` instance is reading
   characters.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 303)**

Unknown directive type "attribute".

```
.. attribute:: shlex.source

   This attribute is ``None`` by default.  If you assign a string to it, that
   string will be recognized as a lexical-level inclusion request similar to the
   ``source`` keyword in various shells.  That is, the immediately following token
   will be opened as a filename and input will be taken from that stream until
   EOF, at which point the :meth:`~io.IOBase.close` method of that stream will be
   called and the input source will again become the original input stream.  Source
   requests may be stacked any number of levels deep.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 314)**

Unknown directive type "attribute".

```
.. attribute:: shlex.debug

   If this attribute is numeric and ``1`` or more, a :class:`~shlex.shlex`
   instance will print verbose progress output on its behavior.  If you need
   to use this, you can read the module source code to learn the details.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 321)**

Unknown directive type "attribute".

```
.. attribute:: shlex.lineno

   Source line number (count of newlines seen so far plus one).
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 326)**

Unknown directive type "attribute".

```
.. attribute:: shlex.token

   The token buffer.  It may be useful to examine this when catching exceptions.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 331)**

Unknown directive type "attribute".

```
.. attribute:: shlex.eof

   Token used to determine end of file. This will be set to the empty string
   (``''``), in non-POSIX mode, and to ``None`` in POSIX mode.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 337)

Unknown directive type "attribute".

```
.. attribute:: shlex.punctuation_chars

   A read-only property. Characters that will be considered punctuation. Runs of
   punctuation characters will be returned as a single token. However, note that no
   semantic validity checking will be performed: for example, '>>>' could be
   returned as a token, even though it may not be recognised as such by shells.

   .. versionadded:: 3.6
```

## Parsing Rules

When operating in non-POSIX mode, :class:`~shlex.shlex` will try to obey to the following rules.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 352); *backlink*

Unknown interpreted text role "class".

- Quote characters are not recognized within words (`Do"Not"Separate` is parsed as the single word `Do"Not"Separate`);
- Escape characters are not recognized;
- Enclosing characters in quotes preserve the literal value of all characters within the quotes;
- Closing quotes separate words (`"Do"Separate` is parsed as `"Do"` and `Separate`);
- If :attr:`~shlex.whitespace_split` is `False`, any character not declared to be a word character, whitespace, or a quote will be returned as a single-character token. If it is `True`, :class:`~shlex.shlex` will only split words in whitespaces;

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 366); *backlink*

  Unknown interpreted text role "attr".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 366); *backlink*

  Unknown interpreted text role "class".

- EOF is signaled with an empty string (`''`);
- It's not possible to parse empty strings, even if quoted.

When operating in POSIX mode, :class:`~shlex.shlex` will try to obey to the following parsing rules.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 375); *backlink*

Unknown interpreted text role "class".

- Quotes are stripped out, and do not separate words (`"Do"Not"Separate"` is parsed as the single word `DoNotSeparate`);
- Non-quoted escape characters (e.g. `'\'`) preserve the literal value of the next character that follows;
- Enclosing characters in quotes which are not part of :attr:`~shlex.escapedquotes` (e.g. `"'"`) preserve the literal value of all characters within the quotes;

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 384); *backlink*

  Unknown interpreted text role "attr".

- Enclosing characters in quotes which are part of :attr:`~shlex.escapedquotes` (e.g. `'"'`) preserves the literal value of all characters within the quotes, with the exception of the characters mentioned in :attr:`~shlex.escape`. The escape characters retain its special meaning only when followed by the quote in use, or the escape character itself. Otherwise the escape character will be considered a normal character.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 388); *backlink*

  Unknown interpreted text role "attr".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 388); *backlink*

  Unknown interpreted text role "attr".

- EOF is signaled with a :const:`None` value;

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 396); *backlink*

  Unknown interpreted text role "const".

- Quoted empty strings (`''`) are allowed.

## Improved Compatibility with Shells

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 405)

Unknown directive type "versionadded".

```
.. versionadded:: 3.6
```

The :class:`shlex` class provides compatibility with the parsing performed by common Unix shells like `bash`, `dash`, and `sh`. To take advantage of this compatibility, specify the `punctuation_chars` argument in the constructor. This defaults to `False`, which preserves pre-3.6 behaviour. However, if it is set to `True`, then parsing of the characters `();<>|&` is changed: any run of these characters is returned as a single token. While this is short of a full parser for shells (which would be out of scope for the standard library, given the multiplicity of shells out there), it does allow you to perform processing of command lines more easily than you could otherwise. To illustrate, you can see the difference in the following snippet:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 407); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 418)

Unknown directive type "doctest".

```
.. doctest::
   :options: +NORMALIZE_WHITESPACE

    >>> import shlex
    >>> text = "a && b; c && d || e; f >'abc'; (def \"ghi\")"
    >>> s = shlex.shlex(text, posix=True)
    >>> s.whitespace_split = True
    >>> list(s)
    ['a', '&&', 'b;', 'c', '&&', 'd', '||', 'e;', 'f', '>abc;', '(def', 'ghi)']
    >>> s = shlex.shlex(text, posix=True, punctuation_chars=True)
    >>> s.whitespace_split = True
    >>> list(s)
    ['a', '&&', 'b', ';', 'c', '&&', 'd', '||', 'e', ';', 'f', '>', 'abc', ';',
    '(', 'def', 'ghi', ')']
```

Of course, tokens will be returned which are not valid for shells, and you'll need to implement your own error checks on the returned tokens.

Instead of passing `True` as the value for the punctuation_chars parameter, you can pass a string with specific characters, which will be used to determine which characters constitute punctuation. For example:

```
>>> import shlex
>>> s = shlex.shlex("a && b || c", punctuation_chars="|")
>>> list(s)
['a', '&', '&', 'b', '||', 'c']
```

**Note**

When `punctuation_chars` is specified, the :attr:`~shlex.wordchars` attribute is augmented with the characters `~-./*?=`. That is because these characters can appear in file names (including wildcards) and command-line arguments (e.g. `--color=auto`). Hence:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 445); *backlink***

Unknown interpreted text role "attr".

```
>>> import shlex
>>> s = shlex.shlex('~/a && b-c --color=auto || d *.py?',
...                 punctuation_chars=True)
>>> list(s)
['~/a', '&&', 'b-c', '--color=auto', '||', 'd', '*.py?']
```

However, to match the shell as closely as possible, it is recommended to always use `posix` and :attr:`~shlex.whitespace_split` when using :attr:`~shlex.punctuation_chars`, which will negate :attr:`~shlex.wordchars` entirely.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 456); *backlink***

Unknown interpreted text role "attr".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 456); *backlink***

Unknown interpreted text role "attr".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 456); *backlink***

Unknown interpreted text role "attr".

For best effect, `punctuation_chars` should be set in conjunction with `posix=True`. (Note that `posix=False` is the default for :class:`~shlex.shlex`.)

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)shlex.rst, line 461); *backlink***

Unknown interpreted text role "class".