:mod:`textwrap` --- Text wrapping and filling

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

Unknown directive type "module".

```
.. module:: textwrap
   :synopsis: Text wrapping and filling
```

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

Unknown directive type "moduleauthor".

```
.. moduleauthor:: Greg Ward <gward@python.net>
```

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

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Greg Ward <gward@python.net>
```

Source code: :source:`Lib/textwrap.py`

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 10); *backlink*

Unknown interpreted text role "source".

The :mod:`textwrap` module provides some convenience functions, as well as :class:`TextWrapper`, the class that does all the work. If you're just wrapping or filling one or two text strings, the convenience functions should be good enough; otherwise, you should use an instance of :class:`TextWrapper` for efficiency.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 14); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 14); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 14); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 20)

Unknown directive type "function".

```
.. function:: wrap(text, width=70, *, initial_indent="", \
                   subsequent_indent="", expand_tabs=True, \
                   replace_whitespace=True, fix_sentence_endings=False, \
                    break_long_words=True, drop_whitespace=True, \
                    break_on_hyphens=True, tabsize=8, max_lines=None, \
                    placeholder=' [...]')

   Wraps the single paragraph in *text* (a string) so every line is at most
   *width* characters long.  Returns a list of output lines, without final
   newlines.

   Optional keyword arguments correspond to the instance attributes of
   :class:`TextWrapper`, documented below.

   See the :meth:`TextWrapper.wrap` method for additional details on how
   :func:`wrap` behaves.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 38)

Unknown directive type "function".

```
.. function:: fill(text, width=70, *, initial_indent="", \
                    subsequent_indent="", expand_tabs=True, \
                    replace_whitespace=True, fix_sentence_endings=False, \
                    break_long_words=True, drop_whitespace=True, \
                    break_on_hyphens=True, tabsize=8, \
                    max_lines=None, placeholder=' [...]')

   Wraps the single paragraph in *text*, and returns a single string containing the
   wrapped paragraph.  :func:`fill` is shorthand for  ::

      "\n".join(wrap(text, ...))

   In particular, :func:`fill` accepts exactly the same keyword arguments as
   :func:`wrap`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 54)

Unknown directive type "function".

```
.. function:: shorten(text, width, *, fix_sentence_endings=False, \
                       break_long_words=True, break_on_hyphens=True, \
                       placeholder=' [...]')

   Collapse and truncate the given *text* to fit in the given *width*.

   First the whitespace in *text* is collapsed (all whitespace is replaced by
   single spaces).  If the result fits in the *width*, it is returned.
   Otherwise, enough words are dropped from the end so that the remaining words
   plus the :attr:`placeholder` fit within :attr:`width`::

      >>> textwrap.shorten("Hello  world!", width=12)
      'Hello world!'
      >>> textwrap.shorten("Hello  world!", width=11)
      'Hello [...]'
      >>> textwrap.shorten("Hello world", width=10, placeholder="...")
      'Hello...'

   Optional keyword arguments correspond to the instance attributes of
   :class:`TextWrapper`, documented below.  Note that the whitespace is
   collapsed before the text is passed to the :class:`TextWrapper` :meth:`fill`
   function, so changing the value of :attr:`.tabsize`, :attr:`.expand_tabs`,
   :attr:`.drop_whitespace`, and :attr:`.replace_whitespace` will have no effect.

   .. versionadded:: 3.4
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 80)

Unknown directive type "function".

```
.. function:: dedent(text)

   Remove any common leading whitespace from every line in *text*.
   This can be used to make triple-quoted strings line up with the left edge of the
   display, while still presenting them in the source code in indented form.

   Note that tabs and spaces are both treated as whitespace, but they are not
   equal: the lines ``"  hello"`` and ``"\thello"`` are considered to have no
   common leading whitespace.

   Lines containing only whitespace are ignored in the input and normalized to a
   single newline character in the output.

   For example::

      def test():
          # end first line with \ to avoid the empty line!
          s = '''\
          hello
            world
          '''
          print(repr(s))          # prints '    hello\n      world\n    '
          print(repr(dedent(s)))  # prints 'hello\n  world\n'
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 106)

Unknown directive type "function".

```
.. function:: indent(text, prefix, predicate=None)

   Add *prefix* to the beginning of selected lines in *text*.

   Lines are separated by calling ``text.splitlines(True)``.

   By default, *prefix* is added to all lines that do not consist
   solely of whitespace (including any line endings).

   For example::

      >>> s = 'hello\n\n \nworld'
      >>> indent(s, '  ')
      '  hello\n\n \n  world'

   The optional *predicate* argument can be used to control which lines
   are indented. For example, it is easy to add *prefix* to even empty
   and whitespace-only lines::

      >>> print(indent(s, '+ ', lambda line: True))
      + hello
      +
      +
      + world

   .. versionadded:: 3.3
```

:func:`wrap`, :func:`fill` and :func:`shorten` work by creating a :class:`TextWrapper` instance and calling a single method on it. That instance is not reused, so for applications that process many text strings using :func:`wrap` and/or :func:`fill`, it may be more efficient to create your own :class:`TextWrapper` object.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 134); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 134); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 134); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 134); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 134); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 134); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 134); *backlink*

Unknown interpreted text role "class".

Text is preferably wrapped on whitespaces and right after the hyphens in hyphenated words; only then will long words be broken if necessary, unless :attr:`TextWrapper.break_long_words` is set to false.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 140); *backlink*

Unknown interpreted text role "attr".

The :class:`TextWrapper` constructor accepts a number of optional keyword arguments. Each keyword argument corresponds to an instance attribute, so for example

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 146); *backlink*

Unknown interpreted text role "class".

```
wrapper = TextWrapper(initial_indent="* ")
```

is the same as

```
wrapper = TextWrapper()
wrapper.initial_indent = "* "
```

You can re-use the same :class:`TextWrapper` object many times, and you can change any of its options through direct assignment to instance attributes between uses.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 157); *backlink*

Unknown interpreted text role "class".

The :class:`TextWrapper` instance attributes (and keyword arguments to the constructor) are as follows:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 161); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)textwrap.rst`, line 165)

Unknown directive type "attribute".

```
.. attribute:: width

   (default: ``70``) The maximum length of wrapped lines.  As long as there
   are no individual words in the input text longer than :attr:`width`,
   :class:`TextWrapper` guarantees that no output line will be longer than
   :attr:`width` characters.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library) textwrap.rst, line 173)

Unknown directive type "attribute".

```
.. attribute:: expand_tabs

   (default: ``True``) If true, then all tab characters in *text* will be
   expanded to spaces using the :meth:`expandtabs` method of *text*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library) textwrap.rst, line 179)

Unknown directive type "attribute".

```
.. attribute:: tabsize

   (default: ``8``) If :attr:`expand_tabs` is true, then all tab characters
   in *text* will be expanded to zero or more spaces, depending on the
   current column and the given tab size.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library) textwrap.rst, line 188)

Unknown directive type "attribute".

```
.. attribute:: replace_whitespace

   (default: ``True``) If true, after tab expansion but before wrapping,
   the :meth:`wrap` method will replace each whitespace character
   with a single space.  The whitespace characters replaced are
   as follows: tab, newline, vertical tab, formfeed, and carriage
   return (``'\t\n\v\f\r'``).

   .. note::

      If :attr:`expand_tabs` is false and :attr:`replace_whitespace` is true,
      each tab character will be replaced by a single space, which is *not*
      the same as tab expansion.

   .. note::

      If :attr:`replace_whitespace` is false, newlines may appear in the
      middle of a line and cause strange output. For this reason, text should
      be split into paragraphs (using :meth:`str.splitlines` or similar)
      which are wrapped separately.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library) textwrap.rst, line 210)

Unknown directive type "attribute".

```
.. attribute:: drop_whitespace

   (default: ``True``) If true, whitespace at the beginning and ending of
   every line (after wrapping but before indenting) is dropped.
   Whitespace at the beginning of the paragraph, however, is not dropped
   if non-whitespace follows it.  If whitespace being dropped takes up an
   entire line, the whole line is dropped.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)textwrap.rst, line 219)

Unknown directive type "attribute".

```
.. attribute:: initial_indent

   (default: ``''``) String that will be prepended to the first line of
   wrapped output.  Counts towards the length of the first line.  The empty
   string is not indented.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)textwrap.rst, line 226)

Unknown directive type "attribute".

```
.. attribute:: subsequent_indent

   (default: ``''``) String that will be prepended to all lines of wrapped
   output except the first.  Counts towards the length of each line except
   the first.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)textwrap.rst, line 233)

Unknown directive type "attribute".

```
.. attribute:: fix_sentence_endings

   (default: ``False``) If true, :class:`TextWrapper` attempts to detect
   sentence endings and ensure that sentences are always separated by exactly
   two spaces.  This is generally desired for text in a monospaced font.
   However, the sentence detection algorithm is imperfect: it assumes that a
   sentence ending consists of a lowercase letter followed by one of ``'.'``,
   ``'!'``, or ``'?'``, possibly followed by one of ``'"'`` or ``"'"``,
   followed by a space.  One problem with this is algorithm is that it is
   unable to detect the difference between "Dr." in ::

      [...] Dr. Frankenstein's monster [...]

   and "Spot." in ::

      [...] See Spot. See Spot run [...]

   :attr:`fix_sentence_endings` is false by default.

   Since the sentence detection algorithm relies on ``string.lowercase`` for
   the definition of "lowercase letter", and a convention of using two spaces
   after a period to separate sentences on the same line, it is specific to
   English-language texts.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)textwrap.rst, line 258)

Unknown directive type "attribute".

```
.. attribute:: break_long_words

   (default: ``True``) If true, then words longer than :attr:`width` will be
   broken in order to ensure that no lines are longer than :attr:`width`.  If
   it is false, long words will not be broken, and some lines may be longer
   than :attr:`width`.  (Long words will be put on a line by themselves, in
   order to minimize the amount by which :attr:`width` is exceeded.)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)textwrap.rst, line 267)

Unknown directive type "attribute".

```
.. attribute:: break_on_hyphens
```

```
    (default: ``True``) If true, wrapping will occur preferably on whitespaces
    and right after hyphens in compound words, as it is customary in English.
    If false, only whitespaces will be considered as potentially good places
    for line breaks, but you need to set :attr:`break_long_words` to false if
    you want truly insecable words.  Default behaviour in previous versions
    was to always allow breaking hyphenated words.
```

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

Unknown directive type "attribute".

```
.. attribute:: max_lines

   (default: ``None``) If not ``None``, then the output will contain at most
   *max_lines* lines, with *placeholder* appearing at the end of the output.

   .. versionadded:: 3.4
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library) textwrap.rst`, line 285)

Unknown directive type "index".

```
.. index:: single: ...; placeholder
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library) textwrap.rst`, line 287)

Unknown directive type "attribute".

```
.. attribute:: placeholder

   (default: ``' [...]'``) String that will appear at the end of the output
   text if it has been truncated.

   .. versionadded:: 3.4
```

:class:`TextWrapper` also provides some public methods, analogous to the module-level convenience functions:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library) textwrap.rst`, line 295); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library) textwrap.rst`, line 298)

Unknown directive type "method".

```
.. method:: wrap(text)

   Wraps the single paragraph in *text* (a string) so every line is at most
   :attr:`width` characters long.  All wrapping options are taken from
   instance attributes of the :class:`TextWrapper` instance.  Returns a list
   of output lines, without final newlines.  If the wrapped output has no
   content, the returned list is empty.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library) textwrap.rst`, line 307)

Unknown directive type "method".

```
.. method:: fill(text)
```

```
        Wraps the single paragraph in *text*, and returns a single string
        containing the wrapped paragraph.
```