:mod:`email.iterators`: Iterators

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

Unknown directive type "module".

```
.. module:: email.iterators
   :synopsis: Iterate over a  message object tree.
```

Source code: :source:`Lib/email/iterators.py`

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.iterators.rst`, line 7); ***backlink***

Unknown interpreted text role "source".

Iterating over a message object tree is fairly easy with the :meth:`Message.walk <email.message.Message.walk>` method. The :mod:`email.iterators` module provides some useful higher level iterations over message object trees.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.iterators.rst`, line 11); ***backlink***

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.iterators.rst`, line 11); ***backlink***

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.iterators.rst`, line 17)

Unknown directive type "function".

```
.. function:: body_line_iterator(msg, decode=False)

   This iterates over all the payloads in all the subparts of *msg*, returning the
   string payloads line-by-line.  It skips over all the subpart headers, and it
   skips over any subpart with a payload that isn't a Python string.  This is
   somewhat equivalent to reading the flat text representation of the message from
   a file using :meth:`~io.TextIOBase.readline`, skipping over all the
   intervening headers.

   Optional *decode* is passed through to :meth:`Message.get_payload
   <email.message.Message.get_payload>`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.iterators.rst`, line 30)

Unknown directive type "function".

```
.. function:: typed_subpart_iterator(msg, maintype='text', subtype=None)

   This iterates over all the subparts of *msg*, returning only those subparts that
   match the MIME type specified by *maintype* and *subtype*.

   Note that *subtype* is optional; if omitted, then subpart MIME type matching is
   done only with the main type.  *maintype* is optional too; it defaults to
   :mimetype:`text`.
   Thus, by default :func:`typed_subpart_iterator` returns each subpart that has a
   MIME type of :mimetype:`text/\*`.
```

The following function has been added as a useful debugging tool. It should *not* be considered part of the supported public interface for the package.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.iterators.rst`, line 46)

Unknown directive type "function".

```
.. function:: _structure(msg, fp=None, level=0, include_default=False)

   Prints an indented representation of the content types of the message object
   structure.  For example:

   .. testsetup::

      import email
      from email.iterators import _structure
      somefile = open('../Lib/test/test_email/data/msg_02.txt')

   .. doctest::

      >>> msg = email.message_from_file(somefile)
      >>> _structure(msg)
      multipart/mixed
          text/plain
          text/plain
          multipart/digest
              message/rfc822
                  text/plain
              message/rfc822
                  text/plain
              message/rfc822
                  text/plain
              message/rfc822
                  text/plain
              message/rfc822
                  text/plain
          text/plain

   .. testcleanup::

      somefile.close()

   Optional *fp* is a file-like object to print the output to.  It must be
   suitable for Python's :func:`print` function.  *level* is used internally.
   *include_default*, if true, prints the default type as well.
```