:mod:`os.path` --- Common pathname manipulations

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

Unknown directive type "module".

```
.. module:: os.path
   :synopsis: Operations on pathnames.
```

Source code: :source:`Lib/posixpath.py` (for POSIX) and :source:`Lib/ntpath.py` (for Windows NT).

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

Unknown interpreted text role "source".

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

Unknown interpreted text role "source".

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

Unknown directive type "index".

```
.. index:: single: path; operations
```

This module implements some useful functions on pathnames. To read or write files see :func:`open`, and for accessing the filesystem see the :mod:`os` module. The path parameters can be passed as either strings, or bytes. Applications are encouraged to represent file names as (Unicode) character strings. Unfortunately, some file names may not be representable as strings on Unix, so applications that need to support arbitrary file names on Unix should use bytes objects to represent path names. Vice versa, using bytes objects cannot represent all file names on Windows (in the standard `mbcs` encoding), hence Windows applications should use string objects to access all files.

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

Unknown interpreted text role "func".

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

Unknown interpreted text role "mod".

Unlike a unix shell, Python does not do any *automatic* path expansions. Functions such as :func:`expanduser` and :func:`expandvars` can be invoked explicitly when an application desires shell-like path expansion. (See also the :mod:`glob` module.)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 25); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 25); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 25); *backlink*

Unknown interpreted text role "mod".

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

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 31)

Unknown directive type "seealso".

```
.. seealso::
   The :mod:`pathlib` module offers high-level path objects.
```

Note

All of these functions accept either only bytes or only string objects as their parameters. The result is an object of the same type, if a path or file name is returned.

Note

Since different operating systems have different path name conventions, there are several versions of this module in the standard library. The :mod:`os.path` module is always the path module suitable for the operating system Python is running on, and therefore usable for local paths. However, you can also import and use the individual modules if you want to manipulate a path that is *always* in one of the different formats. They all have the same interface:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 44); *backlink*

Unknown interpreted text role "mod".

- :mod:`posixpath` for UNIX-style paths

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 52); *backlink*

  Unknown interpreted text role "mod".

- :mod:`ntpath` for Windows paths

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 53); *backlink*

  Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 56)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.8

   :func:`exists`, :func:`lexists`, :func:`isdir`, :func:`isfile`,
   :func:`islink`, and :func:`ismount` now return ``False`` instead of
   raising an exception for paths that contain characters or bytes
   unrepresentable at the OS level.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 64)

Unknown directive type "function".

```
.. function:: abspath(path)

   Return a normalized absolutized version of the pathname *path*. On most
   platforms, this is equivalent to calling the function :func:`normpath` as
   follows: ``normpath(join(os.getcwd(), path))``.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 74)

Unknown directive type "function".

```
.. function:: basename(path)
```

```
   Return the base name of pathname *path*.  This is the second element of the
   pair returned by passing *path* to the function :func:`split`.  Note that
   the result of this function is different
   from the Unix :program:`basename` program; where :program:`basename` for
   ``'/foo/bar/'`` returns ``'bar'``, the :func:`basename` function returns an
   empty string (``''``).

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 87)

Unknown directive type "function".

```
.. function:: commonpath(paths)

   Return the longest common sub-path of each pathname in the sequence
   *paths*.  Raise :exc:`ValueError` if *paths* contain both absolute
   and relative pathnames, the *paths* are on the different drives or
   if *paths* is empty.  Unlike :func:`commonprefix`, this returns a
   valid path.

   .. availability:: Unix, Windows.

   .. versionadded:: 3.5

   .. versionchanged:: 3.6
      Accepts a sequence of :term:`path-like objects <path-like object>`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 103)

Unknown directive type "function".

```
.. function:: commonprefix(list)

   Return the longest path prefix (taken character-by-character) that is a
   prefix of all paths in  *list*.  If *list* is empty, return the empty string
   (``''``).

   .. note::

      This function may return invalid paths because it works a
      character at a time.  To obtain a valid path, see
      :func:`commonpath`.

      ::

        >>> os.path.commonprefix(['/usr/lib', '/usr/local/lib'])
        '/usr/l'

        >>> os.path.commonpath(['/usr/lib', '/usr/local/lib'])
        '/usr'

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 127)

Unknown directive type "function".

```
.. function:: dirname(path)

   Return the directory name of pathname *path*.  This is the first element of
   the pair returned by passing *path* to the function :func:`split`.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 136)

Unknown directive type "function".

```
.. function:: exists(path)

   Return ``True`` if *path* refers to an existing path or an open
   file descriptor.  Returns ``False`` for broken symbolic links.  On
```

```
   some platforms, this function may return ``False`` if permission is
   not granted to execute :func:`os.stat` on the requested file, even
   if the *path* physically exists.

   .. versionchanged:: 3.3
      *path* can now be an integer: ``True`` is returned if it is an
       open file descriptor, ``False`` otherwise.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 152)

Unknown directive type "function".

```
.. function:: lexists(path)

   Return ``True`` if *path* refers to an existing path. Returns ``True`` for
   broken symbolic links.   Equivalent to :func:`exists` on platforms lacking
   :func:`os.lstat`.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

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

Unknown directive type "index".

```
.. index:: single: ~ (tilde); home directory expansion
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 164)

Unknown directive type "function".

```
.. function:: expanduser(path)

   On Unix and Windows, return the argument with an initial component of ``~`` or
   ``~user`` replaced by that *user*'s home directory.

   .. index:: module: pwd

   On Unix, an initial ``~`` is replaced by the environment variable :envvar:`HOME`
   if it is set; otherwise the current user's home directory is looked up in the
   password directory through the built-in module :mod:`pwd`. An initial ``~user``
   is looked up directly in the password directory.

   On Windows, :envvar:`USERPROFILE` will be used if set, otherwise a combination
   of :envvar:`HOMEPATH` and :envvar:`HOMEDRIVE` will be used.  An initial
   ``~user`` is handled by checking that the last directory component of the current
   user's home directory matches :envvar:`USERNAME`, and replacing it if so.

   If the expansion fails or if the path does not begin with a tilde, the path is
   returned unchanged.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.

   .. versionchanged:: 3.8
      No longer uses :envvar:`HOME` on Windows.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 190)

Unknown directive type "index".

```
.. index::
   single: $ (dollar); environment variables expansion
   single: % (percent); environment variables expansion (Windows)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 194)

Unknown directive type "function".

```
.. function:: expandvars(path)

   Return the argument with environment variables expanded.  Substrings of the form
```

```
``$name`` or ``${name}`` are replaced by the value of environment variable
*name*.  Malformed variable names and references to non-existing variables are
left unchanged.

On Windows, ``%name%`` expansions are supported in addition to ``$name`` and
``${name}``.

.. versionchanged:: 3.6
   Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 208)

Unknown directive type "function".

```
.. function:: getatime(path)

   Return the time of last access of *path*.  The return value is a floating point number giving
   the number of seconds since the epoch (see the  :mod:`time` module).  Raise
   :exc:`OSError` if the file does not exist or is inaccessible.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 215)

Unknown directive type "function".

```
.. function:: getmtime(path)

   Return the time of last modification of *path*.  The return value is a floating point number
   giving the number of seconds since the epoch (see the  :mod:`time` module).
   Raise :exc:`OSError` if the file does not exist or is inaccessible.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 225)

Unknown directive type "function".

```
.. function:: getctime(path)

   Return the system's ctime which, on some systems (like Unix) is the time of the
   last metadata change, and, on others (like Windows), is the creation time for *path*.
   The return value is a number giving the number of seconds since the epoch (see
   the  :mod:`time` module).  Raise :exc:`OSError` if the file does not exist or
   is inaccessible.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 237)

Unknown directive type "function".

```
.. function:: getsize(path)

   Return the size, in bytes, of *path*.  Raise :exc:`OSError` if the file does
   not exist or is inaccessible.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 246)

Unknown directive type "function".

```
.. function:: isabs(path)

   Return ``True`` if *path* is an absolute pathname.  On Unix, that means it
   begins with a slash, on Windows that it begins with a (back)slash after chopping
   off a potential drive letter.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 256)

Unknown directive type "function".

```
.. function:: isfile(path)

   Return ``True`` if *path* is an :func:`existing <exists>` regular file.
   This follows symbolic links, so both :func:`islink` and :func:`isfile` can
   be true for the same path.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 266)

Unknown directive type "function".

```
.. function:: isdir(path)

   Return ``True`` if *path* is an :func:`existing <exists>` directory.  This
   follows symbolic links, so both :func:`islink` and :func:`isdir` can be true
   for the same path.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 276)

Unknown directive type "function".

```
.. function:: islink(path)

   Return ``True`` if *path* refers to an :func:`existing <exists>` directory
   entry that is a symbolic link.  Always ``False`` if symbolic links are not
   supported by the Python runtime.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 286)

Unknown directive type "function".

```
.. function:: ismount(path)

   Return ``True`` if pathname *path* is a :dfn:`mount point`: a point in a
   file system where a different file system has been mounted.  On POSIX, the
   function checks whether *path*'s parent, :file:`{path}/..`, is on a different
   device than *path*, or whether :file:`{path}/..` and *path* point to the same
   i-node on the same device --- this should detect mount points for all Unix
   and POSIX variants.  It is not able to reliably detect bind mounts on the
   same filesystem.  On Windows, a drive letter root and a share UNC are
   always mount points, and for any other path ``GetVolumePathName`` is called
   to see if it is different from the input path.

   .. versionadded:: 3.4
      Support for detecting non-root mount points on Windows.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 305)

Unknown directive type "function".

```
.. function:: join(path, *paths)

   Join one or more path components intelligently.  The return value is the
   concatenation of *path* and any members of *\*paths* with exactly one
   directory separator following each non-empty part except the last, meaning
   that the result will only end in a separator if the last part is empty.  If
```

```
a component is an absolute path, all previous components are thrown away
and joining continues from the absolute path component.

On Windows, the drive letter is not reset when an absolute path component
(e.g., ``r'\foo'``) is encountered.  If a component contains a drive
letter, all previous components are thrown away and the drive letter is
reset.  Note that since there is a current directory for each drive,
``os.path.join("c:", "foo")`` represents a path relative to the current
directory on drive :file:`C:` (:file:`c:foo`), not :file:`c:\\foo`.

.. versionchanged:: 3.6
   Accepts a :term:`path-like object` for *path* and *paths*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 325)

Unknown directive type "function".

```
.. function:: normcase(path)

   Normalize the case of a pathname.  On Windows, convert all characters in the
   pathname to lowercase, and also convert forward slashes to backward slashes.
   On other operating systems, return the path unchanged.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 335)

Unknown directive type "function".

```
.. function:: normpath(path)

   Normalize a pathname by collapsing redundant separators and up-level
   references so that ``A//B``, ``A/B/``, ``A/./B`` and ``A/foo/../B`` all
   become ``A/B``.  This string manipulation may change the meaning of a path
   that contains symbolic links.  On Windows, it converts forward slashes to
   backward slashes. To normalize case, use :func:`normcase`.

  .. note::
      On POSIX systems, in accordance with `IEEE Std 1003.1 2013 Edition; 4.13
      Pathname Resolution <http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap04.html#tag_04
      if a pathname begins with exactly two slashes, the first component
      following the leading characters may be interpreted in an implementation-defined
      manner, although more than two leading characters shall be treated as a
      single character.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 355)

Unknown directive type "function".

```
.. function:: realpath(path, *, strict=False)

   Return the canonical path of the specified filename, eliminating any symbolic
   links encountered in the path (if they are supported by the operating
   system).

   If a path doesn't exist or a symlink loop is encountered, and *strict* is
   ``True``, :exc:`OSError` is raised. If *strict* is ``False``, the path is
   resolved as far as possible and any remainder is appended without checking
   whether it exists.

   .. note::
      This function emulates the operating system's procedure for making a path
      canonical, which differs slightly between Windows and UNIX with respect
      to how links and subsequent path components interact.

      Operating system APIs make paths canonical as needed, so it's not
      normally necessary to call this function.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.

   .. versionchanged:: 3.8
      Symbolic links and junctions are now resolved on Windows.

   .. versionchanged:: 3.10
      The *strict* parameter was added.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 384)

Unknown directive type "function".

```
.. function:: relpath(path, start=os.curdir)

   Return a relative filepath to *path* either from the current directory or
   from an optional *start* directory.  This is a path computation:  the
   filesystem is not accessed to confirm the existence or nature of *path* or
   *start*.  On Windows, :exc:`ValueError` is raised when *path* and *start*
   are on different drives.

   *start* defaults to :attr:`os.curdir`.

   .. availability:: Unix, Windows.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 400)

Unknown directive type "function".

```
.. function:: samefile(path1, path2)

   Return ``True`` if both pathname arguments refer to the same file or directory.
   This is determined by the device number and i-node number and raises an
   exception if an :func:`os.stat` call on either pathname fails.

   .. availability:: Unix, Windows.

   .. versionchanged:: 3.2
      Added Windows support.

   .. versionchanged:: 3.4
      Windows now uses the same implementation as all other platforms.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

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

Unknown directive type "function".

```
.. function:: sameopenfile(fp1, fp2)

   Return ``True`` if the file descriptors *fp1* and *fp2* refer to the same file.

   .. availability:: Unix, Windows.

   .. versionchanged:: 3.2
      Added Windows support.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 431)

Unknown directive type "function".

```
.. function:: samestat(stat1, stat2)

   Return ``True`` if the stat tuples *stat1* and *stat2* refer to the same file.
   These structures may have been returned by :func:`os.fstat`,
   :func:`os.lstat`, or :func:`os.stat`.  This function implements the
   underlying comparison used by :func:`samefile` and :func:`sameopenfile`.

   .. availability:: Unix, Windows.

   .. versionchanged:: 3.4
      Added Windows support.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 447)

Unknown directive type "function".

```
.. function:: split(path)

   Split the pathname *path* into a pair, ``(head, tail)`` where *tail* is the
   last pathname component and *head* is everything leading up to that.  The
   *tail* part will never contain a slash; if *path* ends in a slash, *tail*
   will be empty.  If there is no slash in *path*, *head* will be empty.  If
   *path* is empty, both *head* and *tail* are empty.  Trailing slashes are
   stripped from *head* unless it is the root (one or more slashes only).  In
   all cases, ``join(head, tail)`` returns a path to the same location as *path*
   (but the strings may differ).  Also see the functions :func:`dirname` and
   :func:`basename`.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 463)

Unknown directive type "function".

```
.. function:: splitdrive(path)

   Split the pathname *path* into a pair ``(drive, tail)`` where *drive* is either
   a mount point or the empty string.  On systems which do not use drive
   specifications, *drive* will always be the empty string.  In all cases, ``drive
   + tail`` will be the same as *path*.

   On Windows, splits a pathname into drive/UNC sharepoint and relative path.

   If the path contains a drive letter, drive will contain everything
   up to and including the colon::

      >>> splitdrive("c:/dir")
      ("c:", "/dir")

   If the path contains a UNC path, drive will contain the host name
   and share, up to but not including the fourth separator::

      >>> splitdrive("//host/computer/dir")
      ("//host/computer", "/dir")

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 488)

Unknown directive type "function".

```
.. function:: splitext(path)

   Split the pathname *path* into a pair ``(root, ext)``  such that ``root + ext ==
   path``, and the extension, *ext*, is empty or begins with a period and contains at
   most one period.

   If the path contains no extension, *ext* will be ``''``::

      >>> splitext('bar')
      ('bar', '')

   If the path contains an extension, then *ext* will be set to this extension,
   including the leading period. Note that previous periods will be ignored::

      >>> splitext('foo.bar.exe')
      ('foo.bar', '.exe')
      >>> splitext('/foo/bar.exe')
      ('/foo/bar', '.exe')

   Leading periods of the last component of the path are considered to
   be part of the root::

      >>> splitext('.cshrc')
      ('.cshrc', '')
      >>> splitext('/foo/....jpg')
      ('/foo/....jpg', '')

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]os.path.rst, line 519)

Unknown directive type "data".

```
.. data:: supports_unicode_filenames

   ``True`` if arbitrary Unicode strings can be used as file names (within limitations
   imposed by the file system).
```