:mod:`glob` --- Unix style pathname pattern expansion

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

Unknown directive type "module".

```
.. module:: glob
   :synopsis: Unix shell style pathname pattern expansion.
```

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

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

Unknown directive type "index".

```
.. index:: single: filenames; pathname expansion
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.rst, line 13)

Unknown directive type "index".

```
.. index::
   single: * (asterisk); in glob-style wildcards
   single: ? (question mark); in glob-style wildcards
   single: [] (square brackets); in glob-style wildcards
   single: ! (exclamation); in glob-style wildcards
   single: - (minus); in glob-style wildcards
   single: . (dot); in glob-style wildcards
```

The :mod:`glob` module finds all the pathnames matching a specified pattern according to the rules used by the Unix shell, although results are returned in arbitrary order. No tilde expansion is done, but `*`, `?`, and character ranges expressed with `[]` will be correctly matched. This is done by using the :func:`os.scandir` and :func:`fnmatch.fnmatch` functions in concert, and not by actually invoking a subshell. Note that unlike :func:`fnmatch.fnmatch`, :mod:`glob` treats filenames beginning with a dot (`.`) as special cases. (For tilde and shell variable expansion, use :func:`os.path.expanduser` and :func:`os.path.expandvars`.)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.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)glob.rst, line 21); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.rst, line 21); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.rst, line 21); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.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)glob.rst, line 21); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.rst, line 21); *backlink*

Unknown interpreted text role "func".

For a literal match, wrap the meta-characters in brackets. For example, `'[?]'` matches the character `'?'`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.rst, line 35)

Unknown directive type "seealso".

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

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.rst, line 39)

Unknown directive type "function".

```
.. function:: glob(pathname, *, root_dir=None, dir_fd=None, recursive=False, \
                   include_hidden=False)

   Return a possibly-empty list of path names that match *pathname*, which must be
   a string containing a path specification. *pathname* can be either absolute
   (like :file:`/usr/src/Python-1.5/Makefile`) or relative (like
   :file:`../../Tools/\*/\*.gif`), and can contain shell-style wildcards. Broken
   symlinks are included in the results (as in the shell). Whether or not the
   results are sorted depends on the file system.  If a file that satisfies
   conditions is removed or added during the call of this function, whether
   a path name for that file be included is unspecified.

   If *root_dir* is not ``None``, it should be a :term:`path-like object`
   specifying the root directory for searching.  It has the same effect on
   :func:`glob` as changing the current directory before calling it.  If
   *pathname* is relative, the result will contain paths relative to
   *root_dir*.

   This function can support :ref:`paths relative to directory descriptors
   <dir_fd>` with the *dir_fd* parameter.

   .. index::
      single: **; in glob-style wildcards

   If *recursive* is true, the pattern "``**``" will match any files and zero or
   more directories, subdirectories and symbolic links to directories. If the
   pattern is followed by an :data:`os.sep` or :data:`os.altsep` then files will not
   match.

   If *include_hidden* is true, "``**``" pattern will match hidden directories.

   .. audit-event:: glob.glob pathname,recursive glob.glob
   .. audit-event:: glob.glob/2 pathname,recursive,root_dir,dir_fd glob.glob

   .. note::
      Using the "``**``" pattern in large directory trees may consume
      an inordinate amount of time.
```

```
   .. versionchanged:: 3.5
      Support for recursive globs using "``**``".

   .. versionchanged:: 3.10
      Added the *root_dir* and *dir_fd* parameters.

   .. versionchanged:: 3.11
      Added the *include_hidden* parameter.
```

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

Unknown directive type "function".

```
.. function:: iglob(pathname, *, root_dir=None, dir_fd=None, recursive=False, \
                    include_hidden=False)

   Return an :term:`iterator` which yields the same values as :func:`glob`
   without actually storing them all simultaneously.

   .. audit-event:: glob.glob pathname,recursive glob.iglob
   .. audit-event:: glob.glob/2 pathname,recursive,root_dir,dir_fd glob.iglob

   .. versionchanged:: 3.5
      Support for recursive globs using "``**``".

   .. versionchanged:: 3.10
      Added the *root_dir* and *dir_fd* parameters.

   .. versionchanged:: 3.11
      Added the *include_hidden* parameter.
```

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

Unknown directive type "function".

```
.. function:: escape(pathname)

   Escape all special characters (``'?'``, ``'*'`` and ``'['``).
   This is useful if you want to match an arbitrary literal string that may
   have special characters in it.  Special characters in drive/UNC
   sharepoints are not escaped, e.g. on Windows
   ``escape('//?/c:/Quo vadis?.txt')`` returns ``'//?/c:/Quo vadis[?].txt'``.

   .. versionadded:: 3.4
```

For example, consider a directory containing the following files: :file:`1.gif`, :file:`2.txt`, :file:`card.gif` and a subdirectory :file:`sub` which contains only the file :file:`3.txt`. :func:`glob` will produce the following results. Notice how any leading components of the path are preserved.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.rst, line 117); *backlink*

Unknown interpreted text role "file".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.rst, line 117); *backlink*

Unknown interpreted text role "file".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.rst, line 117); *backlink*

Unknown interpreted text role "file".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.rst, line 117); *backlink*

Unknown interpreted text role "file".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.rst, line 117); *backlink*

Unknown interpreted text role "file".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.rst, line 117); *backlink*

Unknown interpreted text role "func".

```
>>> import glob
>>> glob.glob('./[0-9].*')
['./1.gif', './2.txt']
>>> glob.glob('*.gif')
['1.gif', 'card.gif']
>>> glob.glob('?.gif')
['1.gif']
>>> glob.glob('**/*.txt', recursive=True)
['2.txt', 'sub/3.txt']
>>> glob.glob('./**/', recursive=True)
['./', './sub/']
```

If the directory contains files starting with `.` they won't be matched by default. For example, consider a directory containing :file:`card.gif` and :file:`.card.gif`:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.rst, line 135); *backlink*

Unknown interpreted text role "file".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.rst, line 135); *backlink*

Unknown interpreted text role "file".

```
>>> import glob
>>> glob.glob('*.gif')
['card.gif']
>>> glob.glob('.c*')
['.card.gif']
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)glob.rst, line 145)

Unknown directive type "seealso".

```
.. seealso::

   Module :mod:`fnmatch`
      Shell-style filename (not path) expansion
```