:mod:`modulefinder` --- Find modules used by a script

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

Unknown directive type "module".

```
.. module:: modulefinder
   :synopsis: Find modules used by a script.
```

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

Unknown directive type "sectionauthor".

```
.. sectionauthor:: A.M. Kuchling <amk@amk.ca>
```

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

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

Unknown interpreted text role "source".

This module provides a :class:`ModuleFinder` class that can be used to determine the set of modules imported by a script. `modulefinder.py` can also be run as a script, giving the filename of a Python script as its argument, after which a report of the imported modules will be printed.

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

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)modulefinder.rst, line 19)

Unknown directive type "function".

```
.. function:: AddPackagePath(pkg_name, path)

   Record that the package named *pkg_name* can be found in the specified *path*.
```

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

Unknown directive type "function".

```
.. function:: ReplacePackage(oldname, newname)

   Allows specifying that the module named *oldname* is in fact the package named
   *newname*.
```

This class provides :meth:`run_script` and :meth:`report` methods to determine the set of modules imported by a script. *path* can be a list of directories to search for modules; if not specified, `sys.path` is used. *debug* sets the debugging level; higher values make the class print debugging messages about what it's doing. *excludes* is a list of module names to exclude from the analysis. *replace_paths* is a list of `(oldpath, newpath)` tuples that will be replaced in module paths.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)modulefinder.rst, line 32); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)modulefinder.rst, line 32); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)modulefinder.rst, line 41)

Unknown directive type "method".

```
.. method:: report()

   Print a report to standard output that lists the modules imported by the
   script and their paths, as well as modules that are missing or seem to be
   missing.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)modulefinder.rst, line 47)

Unknown directive type "method".

```
.. method:: run_script(pathname)

   Analyze the contents of the *pathname* file, which must contain Python
   code.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)modulefinder.rst, line 52)

Unknown directive type "attribute".

```
.. attribute:: modules

   A dictionary mapping module names to modules. See
   :ref:`modulefinder-example`.
```

Example usage of :class:`ModuleFinder`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)modulefinder.rst, line 60); *backlink*

Unknown interpreted text role "class".

The script that is going to get analyzed later on (bacon.py):

```
import re, itertools

try:
    import baconhameggs
except ImportError:
    pass

try:
    import guido.python.ham
except ImportError:
    pass
```

The script that will output the report of bacon.py:

```
from modulefinder import ModuleFinder

finder = ModuleFinder()
finder.run_script('bacon.py')

print('Loaded modules:')
```

```
    for name, mod in finder.modules.items():
        print('%s: ' % name, end='')
        print(','.join(list(mod.globalnames.keys())[:3]))

    print('-'*50)
    print('Modules not imported:')
    print('\n'.join(finder.badmodules.keys()))
```

Sample output (may vary depending on the architecture):

```
    Loaded modules:
    _types:
    copyreg:  _inverted_registry,_slotnames,__all__
    re._compiler:  isstring,_sre,_optimize_unicode
    _sre:
    re._constants:  REPEAT_ONE,makedict,AT_END_LINE
    sys:
    re:  __module__,finditer,_expand
    itertools:
    __main__:  re,itertools,baconhameggs
    re._parser:  _PATTERNENDERS,SRE_FLAG_UNICODE
    array:
    types:  __module__,IntType,TypeType
    ----------------------------------------------------
    Modules not imported:
    guido.python.ham
    baconhameggs
```