The initialization of the :data:`sys.path` module search path

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]sys_path_init.rst`, line 3); *backlink*
>
> Unknown interpreted text role "data".

A module search path is initialized when Python starts. This module search path may be accessed at :data:`sys.path`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]sys_path_init.rst`, line 6); *backlink*
>
> Unknown interpreted text role "data".

The first entry in the module search path is the directory that contains the input script, if there is one. Otherwise, the first entry is the current directory, which is the case when executing the interactive shell, a :option:`-c` command, or :option:`-m` module.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]sys_path_init.rst`, line 9); *backlink*
>
> Unknown interpreted text role "option".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]sys_path_init.rst`, line 9); *backlink*
>
> Unknown interpreted text role "option".

The :envvar:`PYTHONPATH` environment variable is often used to add directories to the search path. If this environment variable is found then the contents are added to the module search path.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]sys_path_init.rst`, line 14); *backlink*
>
> Unknown interpreted text role "envvar".

> **Note**
>
> :envvar:`PYTHONPATH` will affect all installed Python versions/environments. Be wary of setting this in your shell profile or global environment variables. The :mod:`site` module offers more nuanced techniques as mentioned below.
>
> > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]sys_path_init.rst`, line 20); *backlink*
> >
> > Unknown interpreted text role "envvar".
>
> > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]sys_path_init.rst`, line 20); *backlink*
> >
> > Unknown interpreted text role "mod".

The next items added are the directories containing standard Python modules as well as any :term:`extension module`s that these modules depend on. Extension modules are `.pyd` files on Windows and `.so` files on other platforms. The directory with the platform-independent Python modules is called `prefix`. The directory with the extension modules is called `exec_prefix`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]sys_path_init.rst`, line 24); *backlink*
>
> Unknown interpreted text role "term".

The :envvar:`PYTHONHOME` environment variable may be used to set the `prefix` and `exec_prefix` locations. Otherwise these directories are found by using the Python executable as a starting point and then looking for various 'landmark' files and directories. Note that any symbolic links are followed so the real Python executable location is used as the search starting point. The Python executable location is called `home`.

Once `home` is determined, the `prefix` directory is found by first looking for :file:`python{majorversion}{minorversion}.zip` (`python311.zip`). On Windows the zip archive is searched for in `home` and on Unix the archive is expected to be in :file:`lib`. Note that the expected zip archive location is added to the module search path even if the archive does not exist. If no archive was found, Python on Windows will continue the search for `prefix` by looking for :file:`Lib\\os.py`. Python on Unix will look for :file:`lib/python{majorversion}.{minorversion}/os.py` (`lib/python3.11/os.py`). On Windows `prefix` and `exec_prefix` are the same, however on other platforms :file:`lib/python{majorversion}.{minorversion}/lib-dynload` (`lib/python3.11/lib-dynload`) is searched for and used as an anchor for `exec_prefix`. On some platforms :file:`lib` may be :file:`lib64` or another value, see :data:`sys.platlibdir` and :envvar:`PYTHONPLATLIBDIR`.

Once found, `prefix` and `exec_prefix` are available at :data:`sys.prefix` and :data:`sys.exec_prefix` respectively.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]sys_path_init.rst`, line 50); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]sys_path_init.rst`, line 50); *backlink*

Unknown interpreted text role "data".

Finally, the :mod:`site` module is processed and :file:`site-packages` directories are added to the module search path. A common way to customize the search path is to create :mod:`sitecustomize` or :mod:`usercustomize` modules as described in the :mod:`site` module documentation.

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

Unknown interpreted text role "file".

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

Unknown interpreted text role "mod".

Note

Certain command line options may further affect path calculations. See :option:`-E`, :option:`-I`, :option:`-s` and :option:`-S` for further details.

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

Unknown interpreted text role "option".

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

Unknown interpreted text role "option".

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

Unknown interpreted text role "option".

Virtual environments

If Python is run in a virtual environment (as described at :ref:`tut-venv`) then `prefix` and `exec_prefix` are specific to the virtual environment.

If a `pyvenv.cfg` file is found alongside the main executable, or in the directory one level above the executable, the following variations apply:

- If `home` is an absolute path and :envvar:`PYTHONHOME` is not set, this path is used instead of the path to the main executable when deducing `prefix` and `exec_prefix`.

_pth files

To completely override :data:`sys.path` create a `._pth` file with the same name as the shared library or executable (`python._pth` or `python311._pth`). The shared library path is always known on Windows, however it may not be available on other platforms. In the `._pth` file specify one line for each path to add to :data:`sys.path`. The file based on the shared library name overrides the one based on the executable, which allows paths to be restricted for any program loading the runtime if desired.

When the file exists, all registry and environment variables are ignored, isolated mode is enabled, and :mod:`site` is not imported unless one line in the file specifies `import site`. Blank paths and lines starting with `#` are ignored. Each path may be absolute or relative to the location of the file. Import statements other than to `site` are not permitted, and arbitrary code cannot be specified.

Note that `.pth` files (without leading underscore) will be processed normally by the :mod:`site` module when `import site` has been specified.

Embedded Python

If Python is embedded within another application :c:func:`Py_InitializeFromConfig` and the :c:type:`PyConfig` structure can be used to initialize Python. The path specific details are described at :ref:`init-path-config`. Alternatively the older :c:func:`Py_SetPath` can be used to bypass the initialization of the module search path.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]sys_path_init.rst`, line 100); *backlink*

Unknown interpreted text role "c:func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]sys_path_init.rst`, line 100); *backlink*

Unknown interpreted text role "c:type".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]sys_path_init.rst`, line 100); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]sys_path_init.rst`, line 100); *backlink*

Unknown interpreted text role "c:func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]sys_path_init.rst`, line 105)

Unknown directive type "seealso".

```
.. seealso::

   * :ref:`windows_finding_modules` for detailed Windows notes.
   * :ref:`using-on-unix` for Unix details.
```