# Runtime Features

This chapter covers the features of the Python standard library that allow a program to interact with the interpreter or the environment in which it runs.

During start-up, the interpreter loads the `site` module to configure settings specific to the current installation. The import path is constructed from a combination of environment settings, interpreter build parameters, and configuration files.

The `sys` module is one of the largest in the standard library. It includes functions for accessing a broad range of interpreter and system settings, including interpreter build settings and limits; command line arguments and program exit codes; exception handling; thread debugging and control; the import mechanism and imported modules; runtime control flow tracing; and standard input and output streams for the process.

While sys is focused on interpreter settings, `os` provides access to operating system information. It can be used for portable interfaces to system calls that return details about the running process such as its owner and environment variables. It also includes functions for working with the file system and process management.

Python is often used as a cross-platform language for creating portable programs. Even in a program intended to run anywhere, it is occasionally necessary to know the operating system or hardware architecture of the current system. The `platform` module provides functions to retrieve those settings

The limits for system resources such as the maximum process stack size or number of open files can be probed and changed through the `resource` module. It also reports the current consumption rates, so a process can be monitored for resource leaks.

The `gc` module gives access to the internal state of Python’s garbage collection system. It includes information useful for detecting and breaking object cycles, turning the collector on and off, and adjusting thresholds that automatically trigger collection sweeps.

The `sysconfig` module holds the compile-time variables from the build scripts, and can be used by build and packaging tools to generate paths and other settings dynamically.

ensurepip — Install the Python Package Installer

site — Site-wide Configuration

*This page was last updated 2016-12-31.*