Core Modules

> *"Since the functions in the C runtime library are not part of the Win32 API, we believe the number of applications that will be affected by this bug to be very limited"*
>
> *Microsoft, January 1999*

Overview

Python's standard library covers a wide range of modules. Everything from modules that are as much a part of the Python language as the types and statements defined by the language specification, to obscure modules that are probably useful only to a small number of programs.

This section describes a number of fundamental standard library modules. Any larger Python program is likely to use most of these modules, either directly or indirectly.

Built-in Functions and Exceptions

Two modules are even more basic than all other modules combined: the **__builtin__** module defines built-in functions (like **len**, **int**, and **range**), and the **exceptions** module defines all built-in exceptions.

Python imports both modules when it starts up, and makes their content available for all programs.

Operating System Interface Modules

There are a number of modules providing platform-independent interfaces to the underlying operating system. They are modeled after the POSIX standard API and the standard C library.

The modules in this group include **os**, which provides file and process operations, **os.path** which offers a platform-independent way to pull apart and put together file names, and **time** which provides functions to work with dates and times.

To some extent, networking and thread support modules could also belong in this group, but they are not supported by all Python implementations.

Type Support Modules

Several built-in types have support modules in the standard library. The **string** module implements commonly used string operations, the **math** module provides math operations and constants, and the **cmath** module does the same for complex numbers.

Regular Expressions

The **re** module provides regular expressions support for Python. Regular expressions are string patterns written in a special syntax, which can be used to match strings, and extract substrings.

Language Support Modules

sys gives you access to various interpreter variables, such as the module search path, and the interpreter version. **operator** provides functional equivalents to many built-in operators. **copy** allows you to copy objects. And finally, **gc** gives you more control over the garbage collector facilities in Python 2.0.