

## Navigation

- [index](#)
- [modules](#) |
- [next](#) |
- [previous](#) |
- [Python v2.6.4 documentation](#) »
- [The Python Standard Library](#) »
- [28. Python Runtime Services](#) »

## 28.2. `__builtin__` — Built-in objects ¶

This module provides direct access to all 'built-in' identifiers of Python; for example, `__builtin__.open` is the full name for the built-in function [`open\(\)`](#). See chapter [Built-in Objects](#).

This module is not normally accessed explicitly by most applications, but can be useful in modules that provide objects with the same name as a built-in value, but in which the built-in of that name is also needed. For example, in a module that wants to implement an [`open\(\)`](#) function that wraps the built-in [`open\(\)`](#), this module can be used directly:

```
import __builtin__

def open(path):
    f = __builtin__.open(path, 'r')
    return UpperCaser(f)

class UpperCaser:
    '''Wrapper around a file that converts output to upper-case.'''

    def __init__(self, f):
        self._f = f

    def read(self, count=-1):
        return self._f.read(count).upper()

    # ...
```

**CPython implementation detail:** Most modules have the name `__builtins__` (note the 's') made available as part of their globals. The value of `__builtins__` is normally either this module or the value of this module's `__dict__` attribute. Since this is an implementation detail, it may not be used by alternate implementations of Python.

### Previous topic

[28.1. `sys` — System-specific parameters and functions](#)

### Next topic

[28.3. `future\_builtins` — Python 3 builtins](#)

### This Page

- [Show Source](#)

## Navigation

- [index](#)
- [modules](#) |
- [next](#) |
- [previous](#) |
- [Python v2.6.4 documentation](#) »
- [The Python Standard Library](#) »
- [28. Python Runtime Services](#) »

© [Copyright](#) 1990-2010, Python Software Foundation.

The Python Software Foundation is a non-profit corporation. [Please donate.](#)

Last updated on Feb 26, 2010. Created using [Sphinx](#) 0.6.3.