`!builtins` --- Built-in objects

builtins

This module provides direct access to all 'built-in' identifiers of Python; for example, `builtins.open` is the full name for the built-in function `open`.

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 builtins

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

class UpperCaser:
    '''Wrapper around a file that converts output to uppercase.'''

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

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

    # ...
```

As an implementation detail, most modules have the name `__builtins__` made available as part of their globals. The value of `__builtins__` is normally either this module or the value of this module's `~object.__dict__` attribute. Since this is an implementation detail, it may not be used by alternate implementations of Python.

- `built-in-consts`
- `bltin-exceptions`
- `built-in-funcs`
- `bltin-types`

□□□ #2
□ □□□ □□□ 2026-10-31 17:04:36 UTC
□ □□□ □□□ 2026-10-31 17:04:37 UTC