

附录 II. 魔术属性

表II-1. 基本操作

属性名	语法	说明
<code>__new__</code>	<i>staticmethod</i> <code>object.__new__(cls[, ...])</code>	一个静态方法，用于创建该类的实例。
<code>__init__</code>	<code>object.__init__(self[, ...])</code>	初始化该类的一个实例。
<code>__del__</code>	<code>object.__del__(self)</code>	对象被销毁时自动调用。
<code>__str__</code>	<code>object.__str__(self)</code>	返回对象的非正式字符串表示。
<code>__repr__</code>	<code>object.__repr__(self)</code>	返回对象的正式字符串表示。
<code>__format__</code>	<code>object.__format__(self, format_spec)</code>	返回对象的格式化字符串表示。
<code>__bytes__</code>	<code>object.__bytes__(self)</code>	返回对象的字节串表示。
<code>__lt__</code>	<code>object.__lt__(self, other)</code>	对<表达式求值时自动调用。
<code>__le__</code>	<code>object.__le__(self, other)</code>	对<=表达式求值时自动调用。
<code>__eq__</code>	<code>object.__eq__(self, other)</code>	对==表达式求值时自动调用。
<code>__ne__</code>	<code>object.__ne__(self, other)</code>	对!=表达式求值时自动调用。
<code>__gt__</code>	<code>object.__gt__(self, other)</code>	对>表达式求值时自动调用。
<code>__ge__</code>	<code>object.__ge__(self, other)</code>	对>=表达式求值时自动调用。
<code>__hash__</code>	<code>object.__hash__(self)</code>	返回对象的哈希值。
<code>__bool__</code>	<code>object.__bool__(self)</code>	实现逻辑值检测。
<code>__call__</code>	<code>object.__call__(self[, args...])</code>	使该对象可被调用。

表II-2. 类定义

属性名	语法	说明
<code>__slots__</code>	<code>object.__slots__</code>	指定实例属性的属性名。
<code>__dir__</code>	<code>object.__dir__(self)</code>	返回对象的有效属性列表。
<code>__init_subclass__</code>	<i>classmethod</i> <code>object.__init_subclass__(cls)</code>	一个类方法，当该类被其他类继承时自动调用。
<code>__subclasshook__</code>	<i>classmethod</i> <code>object.__subclasshook__(subclass)</code>	一个类方法，检测指定类是否是该类的子类。
<code>__class_getitem__</code>	<i>classmethod</i> <code>object.__class_getitem__(cls, key)</code>	一个类方法，使该类支持泛型。

表II-3. 反射		
属性名	语法	说明
<code>__getattribute__</code>	<code>object.__getattribute__(self, name)</code>	属性被读取时自动调用，如果属性不存在则抛出AttributeError异常。
<code>__setattr__</code>	<code>object.__setattr__(self, name, value)</code>	属性被设置时自动调用。
<code>__delattr__</code>	<code>object.__delattr__(self, name)</code>	属性被删除时自动调用。
<code>__getattr__</code>	<code>object.__getattr__(self, name)</code>	不存在的属性被读取时自动调用。

表II-4. 描述器		
属性名	语法	说明
<code>__get__</code>	<code>object.__get__(self, instance, owner=None)</code>	通过类属性读取描述器时自动调用，实现描述器的读取控制。
<code>__set__</code>	<code>object.__set__(self, instance, value)</code>	通过类属性设置描述器时自动调用，实现描述器的设置控制。
<code>__delete__</code>	<code>object.__delete__(self, instance)</code>	删除类属性对描述器的引用时自动调用，实现描述器的删除控制。
<code>__set_name__</code>	<code>object.__set_name__(self, owner, name)</code>	创建描述器的所有者类时自动调用，以支持托管属性。

表II-5. 上下文管理器		
属性名	语法	说明
<code>__enter__</code>	<code>object.__enter__(self)</code>	进入关联到此对象的运行时上下文。
<code>__exit__</code>	<code>object.__exit__(self, exc_type, exc_value, traceback)</code>	退出关联到此对象的运行时上下文。

表II-6. 模拟数字类型		
属性名	语法	说明
<code>__int__</code>	<code>object.__int__(self)</code>	返回一个整数。
<code>__float__</code>	<code>object.__float__(self)</code>	返回一个浮点数。
<code>__complex__</code>	<code>object.__complex__(self)</code>	返回一个复数。
<code>__index__</code>	<code>object.__index__(self)</code>	返回一个整型索引。
<code>__add__</code>	<code>object.__add__(self, other)</code>	加法。
<code>__sub__</code>	<code>object.__sub__(self, other)</code>	减法。
<code>__mul__</code>	<code>object.__mul__(self, other)</code>	乘法。
<code>__matmul__</code>	<code>object.__matmul__(self, other)</code>	矩阵乘法。

属性名	语法	说明
<code>__truediv__</code>	<code>object.__truediv__(self, other)</code>	除法。
<code>__floordiv__</code>	<code>object.__floordiv__(self, other)</code>	整除。
<code>__mod__</code>	<code>object.__mod__(self, other)</code>	取余。
<code>__divmod__</code>	<code>object.__divmod__(self, other)</code>	带余除法。
<code>__pow__</code>	<code>object.__pow__(self, other[, modulo])</code>	幂。
<code>__lshift__</code>	<code>object.__lshift__(self, other)</code>	左移。
<code>__rshift__</code>	<code>object.__rshift__(self, other)</code>	右移。
<code>__and__</code>	<code>object.__and__(self, other)</code>	按位与。
<code>__xor__</code>	<code>object.__xor__(self, other)</code>	按位异或。
<code>__or__</code>	<code>object.__or__(self, other)</code>	按位或。
<code>__radd__</code>	<code>object.__radd__(self, other)</code>	反射操作数的加法。
<code>__rsub__</code>	<code>object.__rsub__(self, other)</code>	反射操作数的减法。
<code>__rmul__</code>	<code>object.__rmul__(self, other)</code>	反射操作数的乘法。
<code>__rmatmul__</code>	<code>object.__rmatmul__(self, other)</code>	反射操作数的矩阵乘法。
<code>__rtruediv__</code>	<code>object.__rtruediv__(self, other)</code>	反射操作数的除法。
<code>__rfloordiv__</code>	<code>object.__rfloordiv__(self, other)</code>	反射操作数的整除。
<code>__rmod__</code>	<code>object.__rmod__(self, other)</code>	反射操作数的取余。
<code>__rdivmod__</code>	<code>object.__rdivmod__(self, other)</code>	反射操作数的带余除法。
<code>__rpow__</code>	<code>object.__rpow__(self, other[, modulo])</code>	反射操作数的幂。
<code>__rlshift__</code>	<code>object.__rlshift__(self, other)</code>	反射操作数的左移。
<code>__rrshift__</code>	<code>object.__rrshift__(self, other)</code>	反射操作数的右移。
<code>__rand__</code>	<code>object.__rand__(self, other)</code>	反射操作数的按位与。
<code>__rxor__</code>	<code>object.__rxor__(self, other)</code>	反射操作数的按位异或。
<code>__ror__</code>	<code>object.__ror__(self, other)</code>	反射操作数的按位或。
<code>__iadd__</code>	<code>object.__iadd__(self, other)</code>	扩展赋值加法。
<code>__isub__</code>	<code>object.__isub__(self, other)</code>	扩展赋值减法。
<code>__imul__</code>	<code>object.__imul__(self, other)</code>	扩展赋值乘法。
<code>__imatmul__</code>	<code>object.__imatmul__(self, other)</code>	扩展赋值矩阵乘法。
<code>__itruediv__</code>	<code>object.__itruediv__(self, other)</code>	扩展赋值除法。
<code>__ifloordiv__</code>	<code>object.__ifloordiv__(self, other)</code>	扩展赋值整除。
<code>__imod__</code>	<code>object.__imod__(self, other)</code>	扩展赋值取余。
<code>__ipow__</code>	<code>object.__ipow__(self, other[, modulo])</code>	扩展赋值幂。
<code>__ilshift__</code>	<code>object.__ilshift__(self, other)</code>	扩展赋值左移。
<code>__irshift__</code>	<code>object.__irshift__(self, other)</code>	扩展赋值右移。

属性名	语法	说明
<code>__iand__</code>	<code>object.__iand__(self, other)</code>	扩展赋值按位与。
<code>__ixor__</code>	<code>object.__ixor__(self, other)</code>	扩展赋值按位异或。
<code>__ior__</code>	<code>object.__ior__(self, other)</code>	扩展赋值按位或。
<code>__pos__</code>	<code>object.__pos__(self)</code>	正号。
<code>__neg__</code>	<code>object.__neg__(self)</code>	负号。
<code>__abs__</code>	<code>object.__abs__(self)</code>	绝对值。
<code>__invert__</code>	<code>object.__invert__(self)</code>	按位取反。
<code>__trunc__</code>	<code>object.__trunc__(self)</code>	截断。
<code>__floor__</code>	<code>object.__floor__(self)</code>	向下取整。
<code>__ceil__</code>	<code>object.__ceil__(self)</code>	向上取整。
<code>__round__</code>	<code>object.__round__(self[, ndigits])</code>	舍入。

表II-7. 模拟容器类型

属性名	语法	说明
<code>__len__</code>	<code>object.__len__(self)</code>	返回成员个数。
<code>__length_hint__</code>	<code>object.__length_hint__(self)</code>	返回成员个数的估计值。
<code>__contains__</code>	<code>object.__contains__(self, item)</code>	实现成员检测。
<code>__getitem__</code>	<code>object.__getitem__(self, key)</code>	读取成员时自动调用。
<code>__setitem__</code>	<code>object.__setitem__(self, key, value)</code>	设置成员时自动调用。
<code>__delitem__</code>	<code>object.__delitem__(self, key)</code>	删除成员时自动调用。
<code>__missing__</code>	<code>object.__missing__(self, key)</code>	在找不到键时自动调用。

表II-8. 迭代

属性名	语法	说明
<code>__iter__</code>	<code>object.__iter__(self)</code>	返回一个迭代器。
<code>__reversed__</code>	<code>object.__reversed__(self)</code>	返回一个逆向迭代器。
<code>__next__</code>	<code>object.__next__(self)</code>	返回迭代器的下一项。

表II-9. 模式匹配		
属性名	语法	说明
<code>__match_args__</code>	<code>object.__match_args__</code>	用于支持带位置参数的类模式。

表II-10. 异步操作		
属性名	语法	说明
<code>__await__</code>	<code>object.__await__(self)</code>	返回一个用于实现可等待对象的迭代器。
<code>__aenter__</code>	<code>object.__aenter__(self)</code>	进入关联到此对象的运行时上下文，但返回一个可等待对象。
<code>__aexit__</code>	<code>object.__aexit__(self, exc_type, exc_value, traceback)</code>	退出关联到此对象的运行时上下文，但返回一个可等待对象。
<code>__aiter__</code>	<code>object.__aiter__(self)</code>	返回一个异步迭代器。
<code>__anext__</code>	<code>object.__anext__(self)</code>	返回异步迭代器的下一项。