Common Syntax	Special Method Form
a + b	a.__add__(b); alternatively b.__radd__(a)
a − b	a.__sub__(b); alternatively b.__rsub__(a)
a * b	a.__mul__(b); alternatively b.__rmul__(a)
a / b	a.__truediv__(b); alternatively b.__rtruediv__(a)
a // b	a.__floordiv__(b); alternatively b.__rfloordiv__(a)
a % b	a.__mod__(b); alternatively b.__rmod__(a)
a ** b	a.__pow__(b); alternatively b.__rpow__(a)
a << b	a.__lshift__(b); alternatively b.__rlshift__(a)
a >> b	a.__rshift__(b); alternatively b.__rrshift__(a)
a & b	a.__and__(b); alternatively b.__rand__(a)
a ^ b	a.__xor__(b); alternatively b.__rxor__(a)
a | b	a.__or__(b); alternatively b.__ror__(a)
a += b	a.__iadd__(b)
a −= b	a.__isub__(b)
a *= b	a.__imul__(b)
. . .	. . .
+a	a.__pos__()
−a	a.__neg__()
˜a	a.__invert__()
abs(a)	a.__abs__()
a < b	a.__lt__(b)
a <= b	a.__le__(b)
a > b	a.__gt__(b)
a >= b	a.__ge__(b)
a == b	a.__eq__(b)
a != b	a.__ne__(b)
v in a	a.__contains__(v)
a[k]	a.__getitem__(k)
a[k] = v	a.__setitem__(k,v)
del a[k]	a.__delitem__(k)
a(arg1, arg2, ...)	a.__call__(arg1, arg2, ...)
len(a)	a.__len__()
hash(a)	a.__hash__()
iter(a)	a.__iter__()
next(a)	a.__next__()
bool(a)	a.__bool__()
float(a)	a.__float__()
int(a)	a.__int__()
repr(a)	a.__repr__()
reversed(a)	a.__reversed__()
str(a)	a.__str__()

Table 2.1: Overloaded operations, implemented with Python's special methods.