

Таблица 30.1. Наиболее распространенные методы перегрузки операций

Метод	Что реализует	Для чего вызывается
<code>__init__</code>	Конструктор	Создание объекта: <code>X = Class(args)</code>
<code>__del__</code>	Деструктор	Уничтожение объекта <code>X</code>
<code>__add__</code>	Операция <code>+</code>	<code>X + Y</code> , <code>X += Y</code> , если отсутствует <code>__iadd__</code>
<code>__or__</code>	Операция <code> </code> (побитовое "ИЛИ")	<code>X Y</code> , <code>X = Y</code> , если отсутствует <code>__ior__</code>
<code>__repr__</code> , <code>__str__</code>	Вывод, преобразования	<code>print(X)</code> , <code>repr(X)</code> , <code>str(X)</code>
<code>__call__</code>	Вызовы функций	<code>X(*args, **kwargs)</code>
<code>__getattr__</code>	Извлечение атрибута	<code>X.undefined</code>
<code>__setattr__</code>	Присваивание атрибута	<code>X.any = value</code>
<code>__delattr__</code>	Удаление атрибута	<code>del X.any</code>
<code>__getattribute__</code>	Извлечение атрибута	<code>X.any</code>
<code>__getitem__</code>	Индексирование, нарезание, итерация	<code>X[key]</code> , <code>X[i:j]</code> , циклы <code>for</code> и другие итерационные конструкции, если отсутствует <code>__iter__</code>
<code>__setitem__</code>	Присваивание по индексу и срезу	<code>X[key] = value</code> , <code>X[i:j] = iterable</code>
<code>__delitem__</code>	Удаление по индексу и срезу	<code>del X[key]</code> , <code>del X[i:j]</code>
<code>__len__</code>	Длина	<code>len(X)</code> , проверки истинности, если отсутствует <code>__bool__</code>
<code>__bool__</code>	Булевские проверки	<code>bool(X)</code> , проверки истинности (в Python 2.X называется <code>__nonzero__</code>)
<code>__lt__</code> , <code>__gt__</code> , <code>__le__</code> , <code>__ge__</code> , <code>__eq__</code> , <code>__ne__</code>	Сравнения	<code>X < Y</code> , <code>X > Y</code> , <code>X <= Y</code> , <code>X >= Y</code> , <code>X == Y</code> , <code>X != Y</code> (либо иначе <code>__cmp__</code> только в Python 2.X)
<code>__radd__</code>	Правосторонние операции	<code>Other + X</code>
<code>__iadd__</code>	Дополненные на месте операции	<code>X += Y</code> (либо иначе <code>__add__</code>)
<code>__iter__</code> , <code>__next__</code>	Итерационные контексты	<code>I=iter(X)</code> , <code>next(I)</code> ; циклы <code>for</code> , <code>in</code> , если отсутствует <code>__contains__</code> , все включения, <code>map(F, X)</code> , остальные (<code>__next__</code> в Python 2.X называется <code>next</code>)
<code>__contains__</code>	Проверка членства	<code>item in X</code> (любой итерируемый объект)