

Иногда эти концепции взаимодействуют — например, в `object.X` имя `object` ищется по областям видимости, после чего в результирующих объектах производится поиск имени `X`. Поскольку области видимости и пространства имен жизненно важны для понимания кода на Python, давайте подведем более детальные итоги по правилам.

Простые имена: глобальные, если не выполнено их присваивание

Как уже известно, неуточненные простые имена следуют правилу лексических областей видимости LEGB, кратко описанному при исследовании функций в главе 17 первого тома.

Присваивание ($X = \text{значение}$)

По умолчанию делает имя локальным: создает либо изменяет имя `X` в текущей локальной области видимости, если только оно не объявлено как `global` (или `nonlocal` в Python 3.X).

Ссылка (X)

Ищет имя `X` в текущей локальной области видимости, затем в любых объемлющих функциях, далее в текущей глобальной области видимости, затем во встроенной области видимости согласно правилу LEGB. Поиск во включающих классах не выполняется: взамен имена классов извлекаются как атрибуты объектов.

Кроме того, как упоминалось в главе 17 первого тома, некоторые конструкции в особых сценариях дополнительно локализуют имена (например, переменные в ряде включений и конструкции операторов `try`), но подавляющее большинство имен следуют правилу LEGB.

Имена атрибутов: пространства имен объектов

Мы также видели, что уточненные имена атрибутов ссылаются на атрибуты специфических объектов и подчиняются правилам для модулей и классов. В случае объектов классов и экземпляров правила ссылки дополняются с целью включения процедуры поиска при наследовании.

Присваивание (`object.X = значение`)

Создает или модифицирует имя атрибута `X` в пространстве имен уточняемого объекта `object` и больше нигде. Подъем по дереву наследования происходит только при ссылке на атрибут, но не в случае присваивания значения атрибуту.

Ссылка (`object.X`)

Для объектов, основанных на классах, ищет имя атрибута `X` в `object` и затем во всех доступных классах выше, применяя процедуру поиска при наследовании. Для объектов, не основанных на классах, таких как модули, извлекает `X` из `object` напрямую.

Как упоминалось ранее, предшествующие правила отражают нормальный и типичный сценарий. Правила распознавания имен атрибутов могут варьироваться в классах, которые используют более развитые инструменты, особенно в классах нового стиля — вариант в Python 2.X и стандарт в Python 3.X, которые мы будем исследовать в