

## 11.6 ارجاع های ضعیف (weak references): پایتون دارای سیستم مدیریت خودکار حافظه است (شمارش ارجاع

برای اکثر اشیاء و جمع آوری زباله ([garbage collection](#)) برای حذف چرخه ها). پس از حذف آخرین ارجاع به حافظه، حافظه سریعاً آزاد می شود.

این راهکار برای اکثر برنامه های کاربردی به خوبی کار می کند، اما گاهی نیاز است تا اشیاء را تنها تا زمانی که توسط چیز دیگری در حال استفاده هستند، دنبال کنیم. متأسفانه، فقط دنبال کردن آنها، یک ارجاعی می سازد که آنها را دائمی می کند. ماژول [weakref](#) ابزارهایی برای دنبال کردن اشیاء بدون ایجاد یک ارجاع را فراهم می کند. زمانی که دیگر به شیء احتیاج نداریم، شیء به صورت خودکار از جدول weakref حذف می شود و یک باز فراخوانی (callback) برای اشیاء weakref فعال می شود. برنامه های کاربردی معمول، شامل اشیاء ذخیره سازی (caching) هستند که ساخت آنها گران است.

```
>>> import weakref, gc
>>> class A:
...     def __init__(self, value):
...         self.value = value
...     def __repr__(self):
...         return str(self.value)
...
>>> a = A(10) # create a reference
>>> d = weakref.WeakValueDictionary()
>>> d['primary'] = a # does not create a reference
>>> d['primary'] # fetch the object if it is still alive
10
>>> del a # remove the one reference
>>> gc.collect() # run garbage collection right away
0
>>> d['primary'] # entry was automatically removed
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    d['primary'] # entry was automatically removed
  File "C:/python38/lib/weakref.py", line 46, in __getitem__
    o = self.data[key]()
KeyError: 'primary'
```