Rails模板呈现具有与Ruby想类似的性能特征迭代器。它可以做任何事情，直到你在一个循环里渲染局部模板。对于这个，有两个原因。

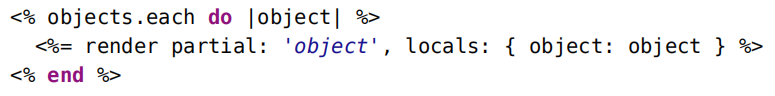
首先，渲染是有代价的。初始化视图对象需要时间，计算执行上下文，并传递所需要的变量。所以，每一个循环中呈现的部分应该是性能不佳的首要原因。

其次，大多数Rails的视图助手都是不安全的迭代器。一个电话不会让你慢下来，但是成千上万个就会让你慢下来。

这两个潜在的性能问题实际上是相同的，在“避免创建其他对象的迭代器”疑问中已经讨论过，在第27页上注意迭代器不安全的Ruby标准函数，在第29页上只需要更高一级的抽象和我们只适用于Rails。所以，详细讨论这些问题，看看我们可以优化什么。

**在循环中更快地呈现局部**

当被要求呈现一组对象时，你的模板代码可能会这样做，看起来是这样的：



这段代码本身没有什么问题，只是在很大的范围内变慢了对象的集合。有多慢呢？我在不同版本的Rails中测量了10,000个空的局部模板，结果并不令人满意。



表2-渲染10000个局部模板

禁用GC后测量，以秒为单位。结果与分位数成线性关系。

虽然10,000个对象不是一个大数据集，但是只呈现占位符，因为使用最新的Rails，他们会让您后退两秒。这是令人不安的。同样令人不安的是，每一个后续版本的Rails的渲染也会变得越来越糟糕。但在你陷入对于Rails 2.x的美好回忆之前，我要指出，即使是0.3秒什么也不做也已经是0.3秒了。Rails 3.0及更高的版本针对这个问题提供了一个名为render collection的解决方案。

