日志之谜 Python 练习

这个练习包含两个部分。你所写的 Python 代码会用到 urllib 库。相关文件在名为 logpuzzle 的文件夹内。

一副动物的图片被纵向切割为许多份图片切片藏与网络中，每一份对应一个 url。这些 url 隐藏于服务器的日志文件中。你的任务是找出那些 url,下载图片切片并还原原始的图片。

这些 url 被储存在 apache 的日志文件中（开源的 apache 服务器为最为广泛使用的网络服务器）。一个日志文件都来源与一个服务器，那些包含想要切片的 url 被藏在这日志中。日志文件通过以下方式编码服务器来源：animal_code.google.com 日志文件的服务器来源为 code.google.com（服务器名称为第一个下划线后面的部分）。名为 animial_code.google.com 的日志文件包含了用于还原那张动物图片的数据。尽管日志文件中的格式和真实的 apache 网络服务器一致，但是除了用于还原图片所需要的数据外，其他的都是从真实的日志文件中随机生成的。

以下是日志文件中一行的样子（现实中 apache 的日志文件就是这个样子）

```
10.254.254.28 - - [06/Aug/2007:00:14:08 -0700] "GET /foo/talks/
HTTP/1.1"

200 5910 "-" "Mozilla/5.0 (X11; U; Linux i686 (x86_64); en-US;
rv:1.8.1.4) Gecko/20070515 Firefox/2.0.0.4"
```

前几个数为发起请求的浏览器的地址。我们最感兴趣的部分是“Get *path* HTTP”，它表明了服务器接收到了网页请求的具体路径。路径（*path*）本身并不带空格，但是由空格与 Get 和 HTTP 分开（正则表达式建议：\S （大写 S）用于匹配所有非空格字符）。在某些行中，路径部分会包含 “puzzle”，找出所有这样的行并忽略其他的。

A 部分，由日志到 URL

完成 read_urls(filename)函数，该函数从日志文件中提取目标 url。在日志文件中找出所有的包含"puzzle"路径的 url，再从日志名提取服务器名称，将二者结合得到完整的 url。如"http://www.example.com/path/puzzle/from/inside/file"。将重复的 url 去

掉。read_urls()函数应该返还一个包含完整 url 的列表，列表内 url 按照字母顺序排列并且无重复。按字母顺序排列的 url 能够由左到右返还正确的图片片段，帮助我们正确重建原始图片。在最简便的方案中，main()只需要逐行打印每一个 url。

```
$ ./logpuzzle.py animal_code.google.com

http://code.google.com/something/puzzle-animal-baaa.jpg

http://code.google.com/something/puzzle-animal-baab.jpg

...
```

B 部分：下载图片

完成 download_images()函数，该函数输入变量为一个排好序的 url 列表和一个地址。将 url 中的图片下载到对应地址中，如果地址不存在，你的函数应该自动创建（用 os 库来创建地址，用 urllib.urlretrieve()来下载 url）。用一个简单的方案来命名你下载的图片，如"img0", "img1", "img2"等。因为下载图片会需要一些时间，你可能会想打印“下载中”的字样来提示你的程序正在工作中。如何将被切分的竖条图片切片很好的还原成原始图片呢？你可以通过一个 html 文件来做到（不需要 HTML 知识）。

download_images()函数还应该在目录中新建一个 index.html 文件。在 html 中用 *img* tag 来展示所有下载的图片切片。所有切片的 img tag 应该无间隔的放在同一行。这样浏览器会自动无缝显示所有切片。你不需要懂得 HTML 来完成这一步，只需要按如下格式编写你的 html 文件。

```
<verbatim>

<html>

<body>

<img src="/edu/python/exercises/img0"><img
src="/edu/python/exercises/img1"><img
src="/edu/python/exercises/img2">...

</body>
```

```
</html>
```

当你完成这个问题下载工作以后，你的文件夹应该有以下内容。

```
$ ./logpuzzle.py --todir animaldir animal_code.google.com

$ ls animaldir

img0  img1  img2  img3  img4  img5  img6  img7  img8  img9  index.html
```

如果一切顺利，在浏览器打开 index.html，你应该能看到动物的原始图片。这是个什么动物呢？

C 部分：图片解码

第二个部分包含了一张著名地点的图片。但是为了看到它，你不能像完成第一个任务那样将全部 url 按字幕顺序来排列。你应该这样做：如果 url 的结尾是"-字符串-字符串.jpg"格式，例如 http://example.com/foo/puzzle/bar-abab-baaa.jpg，那用于排序的字符是第二组字符，即 baaa。所以在这个部分中你需要根据第二组字符来将你的 url 按字母表顺序排列。

改变你的代码让它正确的将 url 按要求排序，然后你应该就能够解码第二张秘密照片。照片里是哪？