

urlretrieve 方法的使用

`urlretrieve(url[, filename[, reporthook[, data]]])`:

`filename` 指定保存到本地的路径 (若未指定该, `urlretrieve` 生成一个临时文件保存数据)

`reporthook` 回调函数, 当连接上服务器、以及相应的数据块传输完毕的时候会触发该回调

`data` 指 post 到服务器的数据 `urlretrieve(url[, filename[, reporthook[, data]]])` 此函数将远程的数据下载到本地, 其中参数 `url` 是 URL 字符串, `filename` 指定数据保存到本地的路径;

`reporthook(block, size, total)` 是一个回调函数当链接上服务器, 以及相应的数据传送完毕时会触发该函数 (即每下载一块(block)就调用一次回调函数), 我们可以用该函数来显示当前数据下载的进度, 也可以用来限速, `data` 是指 post 到服务器的数据; 该方法返回一个包含两个元素的元组(filename, headers), `filename` 表示保存到本地的路径, `header` 表示服务器的响应头。

```
from __future__ import unicode_literals
```

```
import urllib
```

```
import re
```

```
def callback_f(downloaded_size, block_size, remote_total_size):
```

```
    per = 100.0 * downloaded_size * block_size / remote_total_size
```

```
    if per > 100:    per = 100
```

```
    print("%.2f%%" % per) def getHtml(url):
```

```
    page=urllib.urlopen(url)
```

```
    html=page.read()
```

```
    page.close()
```

```
    return html def getImg(html):
```

```
    imgre=re.compile(r'<img\s.*?src\s*=\s*["\']?([^\s"]+).*?>', re.I)
```

```
    imglist=imgre.findall(html)
```

```
    i=0
```

```
    for imgurl in imglist:
```

```
        if not imgurl.find("http://"):
```

```
            urllib.urlretrieve(imgurl, "img/%d.jpg" % (i), callback_f)
```

```
            i+=1
```

```
    print imgurl ht=getHtml(r"http://www.cnbeta.com/articles/234303.htm")
```

```
    getImg(ht)
```