urllib.request

--urllib.request.urlretrieve(url, filename=None, reporthook=None, data=None)
下载东西，命名为filename带格式

--urllib.request.Request(url,data=None,headers={})
data官方文档：

data must be a bytes object specifying additional data to send to the server, or **None** if no such data is needed. Currently HTTP requests are the only ones that use *data*; the HTTP request will be a POST instead of a GET when the *data* parameter is provided. *data* should be a buffer in the standard *application/x-www-form-urlencoded* format. The **urllib.parse.urlencode()** function takes a mapping or sequence of 2-tuples and returns an ASCII string in this format. It should be encoded to bytes before being used as the *data* parameter.

FE：

```
    data={}
    data['type']='AUTO'
    data['i']=words
    data['doctype']='json'
    data['xmlVersion']='1.8'
    data['keyfrom']='fanyi.web'
    data['ue']='UTF-8'
    data['action']='FY_BY_CLICKBUTTON'
    data['typoResult']='true'

    data=urllib.parse.urlencode(data).encode('utf-8')
```

headers官方文档：

headers should be a dictionary, and will be treated as if **add_header()** was called with each key and value as arguments. This is often used to “spoof” the User-Agent header value, which is used by a browser to identify itself – some HTTP servers only allow requests coming from common browsers as opposed to scripts. For example, Mozilla Firefox may identify itself as "Mozilla/5.0 (X11; U; Linux i686) Gecko/20071127 Firefox/2.0.0.11", while urllib ‘s default user agent string is "Python-urllib/2.6" (on Python 2.6).

FE:

```
head={}
head['User-Agent']='Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.101 Safari/537.36'
```

或者:

FE:

```
import urllib.request
req = urllib.request.Request('http://www.example.com/')
req.add_header('Referer', 'http://www.python.org/')
# Customize the default User-Agent header value:
req.add_header('User-Agent', 'urllib-example/0.1 (Contact: . . .)')
```

--request.add_header(key,value)

--request.set_proxy(host,type)
设置代理服务器

--urllib.request.urlopen(url,data=None[,timeout])
url可以是网址/Request对象
FE：

```
response=urllib.request.urlopen(request)
```

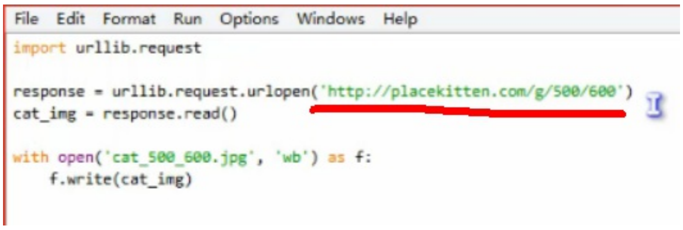
--response.read()

解码网页代码，返回bytes-like

需要decode(’utf-8’)进行解码

FE:

读入在线图片地址进行保存时，文件也是’wb’二进制写入



```
import urllib.request

response = urllib.request.urlopen('http://placekitten.com/g/500/600')
cat_img = response.read()

with open('cat_500_600.jpg', 'wb') as f:
    f.write(cat_img)
```

--urllib.request.ProxyHandler({’type’:’ip:port’})

设置代理

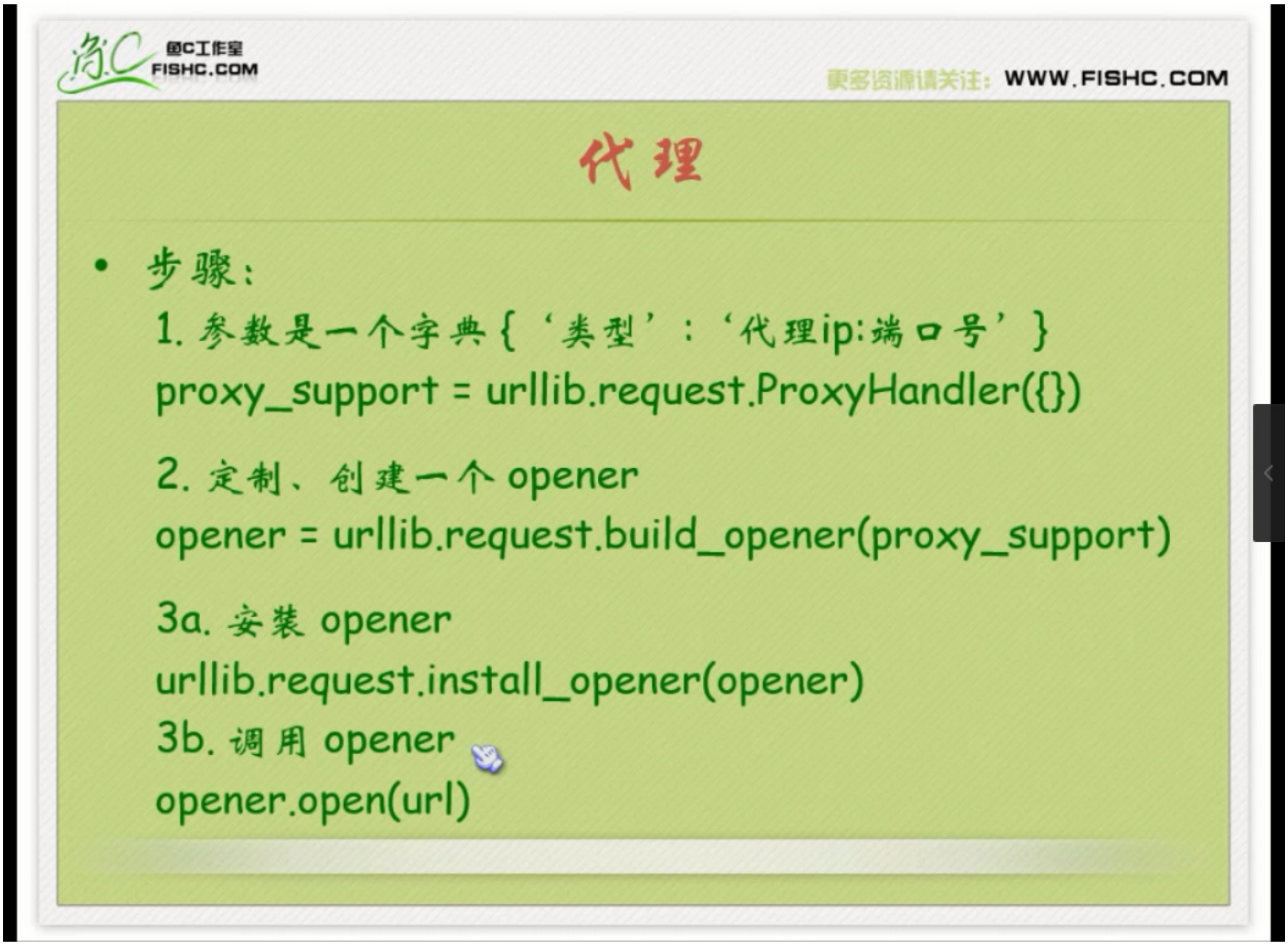
--urllib.request.build_opener(onesupport)

创建opener

--urllib.request.install_opener(oneopener)

安装opener

FE:



代理

- 步骤：

1. 参数是一个字典{‘类型’：‘代理ip:端口号’}

proxy_support = urllib.request.ProxyHandler({})

2. 定制、创建一个 opener

opener = urllib.request.build_opener(proxy_support)

3a. 安装 opener

urllib.request.install_opener(opener)

3b. 调用 opener

opener.open(url)

--opener.open(url)

之前的方法是对于所有的主程序适用

要是只是需要特殊情况使用，则使用这个方法，用urllib.request.set_proxy(ip,type)也行