Python 3 の urllib

urllib.request

Python 3 の urllib では、オブジェクトやメソッドが定義されている名前空間が urllib ではなくて、urllib.request 以下になった。

```
import urllib.request

opener = urllib.request.build_opener()
response = opener.open("http://www.foobar.com")
print(response.read())
```

urllib.request.build_opener([handlers...,]) は、urllib.request.OpenerDirector のインスタンスを返す。
これは、さまざまな HTTP クライアント機能を持ち、open メソッドを持ったオブジェクトである（機能は引数にハンドラオブジェクトを渡して作成する）。

urllib.request.OpenerDirector.open(url, [data, timeout]) は、urllib2.urlopen(url, [data, timeout]) と同じ引数を取る。キーワード引数でも指定可能。
したがって、違和感なく使える。
urllib.request.OpenerDirector.open(...) が返すのは、http.client.HTTPResponse オブジェクト。
これはファイルハンドラのようにしてコンテンツを読み出せる。

Request / Response

URL Opener はデフォルトのものが用意されている。
もっと直感的に HTTP の Request / Response を実現するには、Request オブジェクトを使う。

```
import urllib.request

url_request = urllib.request.Request(url="http://www.yahoo.co.jp")
http_response = urllib.request.urlopen(url_request)
binary_contents = http_response.read()
```

POST はこんな感じ。

```
import urllib.request

PostHeaders = {"Content-Type" : "application/octet-stream"}
post_data_handle = open("C:\post.jpg", "rb")
    post_request          =       urllib.request.Request(url="http://post.intranet.com/image",
data=post_data_handle.read(), method="POST", headers=PostHeaders)
post_response = urllib.request.urlopen(post_request)
```

charset を得ようとする場合

```
import urllib.request

content_charset = ""
url_request = urllib.request.Request(url="http://www.yahoo.co.jp")
http_response = urllib.request.urlopen(url_request)
content_charset = http_response.info().get_content_charset()
if content_charset:
    print(http_response.read().decode(content_charset))
```

http.client.HTTPResponse.info() が返すのは、http.client.HTTPMessage オブジェクト。これを使って、戻ってきた HTTP ヘッダ内の情報を取り出せる。
http.client.HTTPMessage.get_charsets() は charset を表す文字列のリストを返す。HTTP の場合は、取得したページの charset として 0 番目の値だけを参照すればよい、とのこと（メール等のマルチパート MIME に対応できるように、リストで返す仕様にしてあるらしい）。
charset が無い場合には [None]（唯一の値が None であるリスト）が帰ってくるので注意。リストの要素が文字列だという前提でコーディングしていると、例外が発生する。

proxy を使いたい場合

ハンドラオブジェクトを作って渡す。
カスタムのハンドラオブジェクトを使いたい場合には、build_opener を使って、カスタムの URL Opener を作成する必要がある。

```
import urllib.request

opener = urllib.request.build_opener(urllib.request.ProxyHandler({"http": "192.168.1.16:8080"}))
response = opener.open("http://www.foobar.com", timeout=10)
```

こりゃ簡単だ。

2.6 向けの旧記事

最近の urlopen では、タイムアウト値を設定できるようになったようだ。
少なくとも 2.6 の urllib2 ではできるようになっている。

urlopen(url[, data][, timeout])

data は、リクエスト以外に追加でサーバーに送りたい文字列がある場合、とのこと。
timeout は秒数で指定する。

data は普通指定しない。その場合は None でいいので、以下のようになる。

```
import urllib2

result = urllib2.urlopen("http://www.yahoo.co.jp", None, 30)
```