

第十八章 利用樣版系統編排

上一章中建立了留言板的應用程式，接著進入後台進行管理，可是還沒有提及如何做出供他人瀏覽的網頁。我們的資料越來越多，該如何做出網頁呢？

第十六章我們做網頁的方法是直接把HTML寫進字串中，然後利用網址改變以及抓取電腦的現在時間，直接以變數提供給HTML的字串，最後作為HttpResponse()的參數，使之輸出到瀏覽器中。

這是利用Django做網頁的方法之一，當要放進網頁的內容越來越多，同時也希望進行更多的版面配置，使網頁呈現出美觀的外在，把HTML寫進字串的方法就顯得冗贅繁複。其實，這並不是Django做網頁唯一途徑，普遍的作法是利用**樣版系統**編排網頁的版面。

樣版系統結合HTML語法，輔以**標籤及過濾器**，結合程式語言的部份功能，在Django中作為描述網頁編排的標記語言。這使Django相較其他環繞MVC原則的網頁框架有了一個明顯的特色，也就是Django除了遵守MVC原則外，另具備MTV的概念。

MTV的概念

MTV為Model-Template-View的頭字母縮寫詞，M與MVC原則中的M的意義一樣，V稍有差異，MTV的V為控制網頁顯示的方法。T為Template的第一個字母，中文意思就是樣版，樣版系統正是我們這一章的主題。

MTV可以直接與專案內的檔案連起來思考，M如同留言板應用程式中的models.py，V為views.py。至於T，因為一個樣版就是單一個檔案，專案中的樣版數沒有限制，這是說網頁的顯示依需要可以套用不同的樣版，實際上我們會建立一個名為「template」的資料夾放置樣版的檔案。

為什麼前面要介紹MVC，這邊要介紹MTV呢？兩者的內涵似乎重疊了許多？因為MVC是一般所倡導的概念，然而就Django而言，MVC的V是MTV中的T，MVC的C則是MTV中的V，兩者的概念說到底是一樣的。

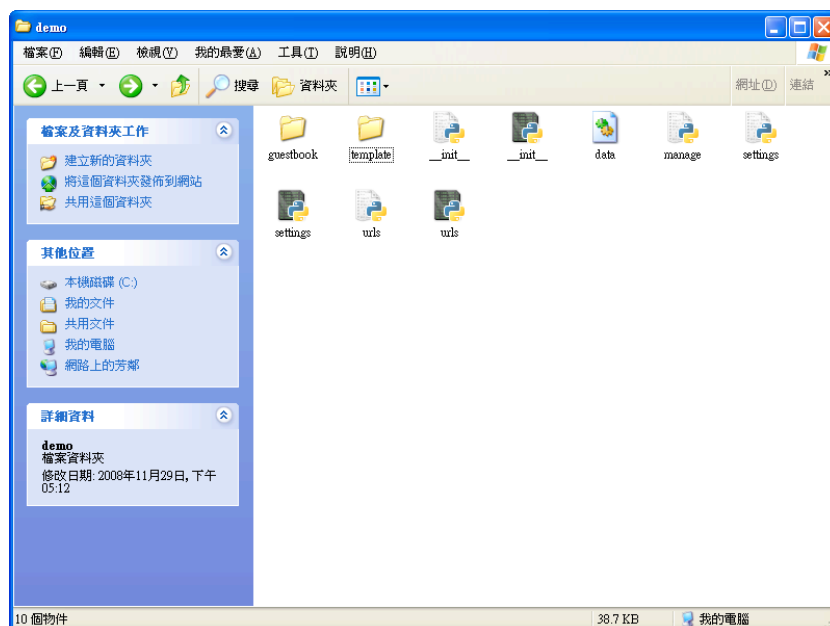
原因不外由於MVC是很常用的詞，而MTV為針對解釋Django之情形所用的詞，MVC在Django之中容易誤解，尤其在Controller的部份，廣義的說整個Django都算是屬於Controller的部份，然而MTV就不會了，因為MTV各有專屬的檔案或資料夾。

我們製作樣版檔案會用到Django的樣版語言，這當中也需要一些HTML，這似乎是要進入另一個與Python完全不同的主體嗎？不是的，樣版系統作為網頁顯示的元件，其實這也是基於Django的設計哲學，「寬鬆的結合」，MTV每個元件各自分開獨立運作。

接著繼續下去我們會發現樣版語言跟Python頗為相似，不過還是得留心不同的地方。

now.html及page.html

我們延續上一章的demo專案，在demo資料夾內建立一個template資料夾。



然後我們先以第十六章「現在時間是 %s。」及「您來到了第 %s 頁；」的例子，簡介樣版系統的語法。樣版的檔案實際上就是HTML檔案，我們先在template資料夾中建立now.html，作為顯示現在時間的樣版，裡頭放入以下的內容。

```
<html>
<head>
  <title>顯示現在時間</title>
</head>

<body>
  <h1>現在時間是 {{ current_time }} 。</h1>

  你好啊！
  右邊會隨機出現一個0到9之間的整數： {{ number|random }}
</body>
</html>
```

其中，兩個大括弧中間包著的current_time，

```
{{ current_time }}
```

 這在樣版中表示current_time為一個變數。而

```
{{ number|random }}
```

 number也是變數，其後的「|」為管線命令符號，經過random參數的隨機透析，使number變數中的數值以過濾後的方式呈現，這就是上述的**過濾器**。稍後，我們會抓取電腦現在時間儲存到current_time內，另外會指定一組數字給number。

另外在template資料夾中建立page.html，作為顯示頁數的樣版，內容如下。

```
<html>
<head>
  <title>顯示頁數及現在時間</title>
</head>

<body>
  <h1>現在時間是 {{ current_time }} 。</h1>
  {% ifequal offset "99" %}
    <p>這是最後一頁囉！</p>
  {% else %}
    <p>您來到了第 {{ offset }} 頁。</p>
  {% endifequal %}
</body>
```

```
</html>
```

變數是用兩個大括弧圍起來，而這裡

```
{% ifequal offset "99" %}
```

到

```
{% endifequal %}
```

之間被稱為區塊，大括弧接百分比符號中間的`{% ifequal %}`則是**標籤**，標籤類似Python中的關鍵字，用於指揮程式進行的過程。這裡的`{% ifequal %}`後面接兩個參數，第一個為變數，第二個為數值，判斷其後所接的變數是否與數值相等，若是相等則在網頁中顯示區塊內的

```
<p>這是最後一頁囉！</p>
```

若是不相等則在網頁中顯示「else」標籤中的

```
<p>您來到了第 {{ offset }} 頁。</p>
```

記不記得在第十六章，這部份是寫在view.py的page_counter()函數之中，因為那時候網頁顯示的工作一併交給了view.py的page_counter()函數，現在的分開處理，正是Django設計哲學中「寬鬆的結合」最佳的例證。

有一點需要留意的，樣版語言的標籤與HTML類似，兩者都是成對出現的，如上例`{% ifequal %}`的標籤，標籤有效範圍終止的地方是用`{% endifequal %}`標籤，這也就代表了區塊的結束。

樣版目錄的設定

我們還需要做幾個設定調整才能夠讓網頁顯示，首先開啟setting.py，找到如下的部份。

```
TEMPLATE_DIRS = (
    # Put strings here, like "/home/html/django_templates" or \
    # "C:/www/django/templates".
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative paths.
)
```

這些注解化的文字大略是說需要用字串的格式，將樣版目錄的路徑放置在這裡。我們簡單一點，將這部份的程式碼更改如下。

```
TEMPLATE_DIRS = (
    '.',
)
```

我們所加入的「`'.'`」，好比是將尋找樣版資料夾的根目錄設定為與專案相同。接著將guestbook資料夾中的views.py更改如下。

```
from django.shortcuts import render_to_response
import datetime

def current_datetime(request):
    cdt = datetime.datetime.now()
    now_template = r"template\now.html"
    now_context = {'current_time':cdt, 'number':range(10)}
    return render_to_response(now_template, now_context)

def page_counter(request, offset):
    cdt = datetime.datetime.now()
    page_template = r"template\page.html"
    page_context = {'current_time':cdt, 'offset':offset}
    return render_to_response(page_template, page_context)
```

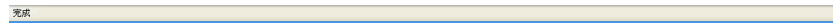
這裡比較特別的是從django的shortcuts模組引入render_to_response()函數，需要兩個參數，第一個參數為指定樣版的路徑，儲存在字串之中，第二個參數則是樣版中變數與變數所儲存的數值，利用配對型態的字典來儲存，key為變數名稱，value則為變數內容，要留意需要用字串型態儲存用作變數名稱的key。

render_to_response()函數是一種捷徑函數，因為利用樣版顯示網頁，實際上需要先把樣版檔案轉換成Template物件，而變數名稱及內容則要轉換為Context物件，捷徑函數便利我們省卻了許多麻煩。

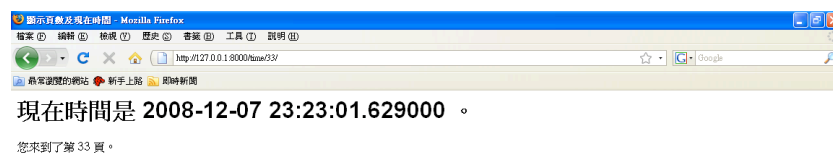
最後還須修改一下urls.py，我們替urlpatterns加入兩行的網址設定。

```
urlpatterns = patterns('',
    (r'^admin/(.*)', admin.site.root),
    (r'^time/$', 'demo.guestbook.views.current_datetime'),
    (r'^time/(\d{1,2})/$', 'demo.guestbook.views.page_counter'),
)
```

現在可以來看看結果了，別忘記要啟動伺服器，然後連結到<http://127.0.0.1:8000/time/>。



再連結到其中的一頁看看吧！



其他的標籤及過濾器

樣版系統除了支援HTML的註解標籤外，本身也有用作註解的標籤。

```
{# 請將註解放在這裡 #}
```

大括弧加井字號，這就是Django樣版語言所用的註解。除了註解之外，Python中用於條件判斷的if陳述，樣版語言也有相對應的標籤。

and、or、not也都支援可用，供比較測試多個變數。別忘了{% if %}標籤要以{% endif %}結尾。當然我們已經見過的{% else %}標籤也有支援，然而elif陳述沒有支援，所以當選擇判斷的條件多於兩個時，就要用巢狀的方式。

```
{% if a and b %}
    變數a與b都為真的話，就會出現這些內容.....
{% else %}
    {% if c %}
        變數a與b都不為真，但是c為真，就會出現這些內容.....
    {% else %}
        變數a、b、c都不為真，就會出現這些內容.....
    {% endif %}
{% endif %}
```

就跟HTML標籤同理，任一個「if」標籤都要以一個「endif」結尾，不然Django會不知道到哪裡結束「if」，從而導致發生錯誤。

另外也有{% for %}標籤。

```
{# 這會依序顯示data中的資料... #}
{% for i in data %}
    {{ i }}<BR />
{% endif %}
```

這會一一列出的在網頁中顯示變數data所儲存的個別數值。

過濾器是針對變數運用的，我們多舉幾個例子說明，譬如變數one中儲存整數1

```
1+1={{ one|add:"1" }}
```

add會替變數加上其後的參數，於是顯示的結果會是「1+1=2」。又如果變數name存放英文姓名

```
{{ name|capfirst }}
```

不論name中字母的大小寫，capfirst都會將第一個字母設定為大寫。我們上面的變數current_time，也可以利用過濾器提出個別的部分

```
{{ current_time|date:"M" }}
```

date用大寫的「M」作為參數，網頁上只會顯示月份。當然，過濾器還有很多，我們再舉一個例子，如變數pi中所儲存的是圓周率3.14159.....

```
{{ pi|floatformat:3 }}
```

floatformat會調整pi在小數點後顯示的位數，這裡給了「3」作為參數，就是會在網頁中顯示圓周率為「3.141」。

Note

Django樣版系統的標籤及過濾器還有很多，可以參考[Built-in template tags and filters](#)。

樣版的繼承

不論在now.html或是page.html中，我們都重複寫了<html></html>、<head></head>及<body></body>等HTML標籤，另外也重複寫了「<h1>現在時間是 {{ current_time }}。</h1>」這一

行。雖然目前例子很簡單，但是當網頁的內容頁數與日俱增，而每一頁又有許多相同的元件時，一再的重複，就顯得冗長累贅。

Python語言的繼承，讓某一型態可以輕易的獲取原先定義型態的屬性與方法，進而覆寫或是增加新的屬性。Django的樣版語言沿襲Python語言繼承的優點，我們可以把網頁的共同元件寫進一個基礎樣版中，不同類型的網頁可以繼承基礎樣版的內容。如下我們以base.html為基礎樣版。

```
<html>
<head>
    <title>{% block title %} 基礎樣版 {% endblock %}</title>
</head>

<body>
    <h1>現在時間是 {{ current_time }} 。</h1>

    {% block content %}{% endblock %}
</body>
</html>
```

標籤「block」通常會出現在某特定區域，如<title></title>這對HTML標籤之內，其後也需要接一個專屬的名稱，作為識別位置之用，然後在被繼承的樣版中，相同的「block」標籤名稱即可覆寫基礎樣版的內容。

我們另外寫now2.html與page2.html來繼承base.html，now2.html如下。

```
{% extends "template\base.html" %}

{% block title %}
    顯示現在時間
{% endblock %}

{% block content %}
    你好啊！
    右邊會隨機出現一個0到9之間的整數： {{ number|random }}
{% endblock %}
```

凡是需要繼承基礎樣版的樣版，當案開頭都要用標籤「extends」，其後接一個字串內含基礎樣版的路徑，然後將所欲覆寫的標籤內容補上，如此一來繼承工作就可順利進行。

省掉了重複撰寫的東西，是否單純了許多呢？page2.html如下。

```
{% extends "template\base.html" %}

{% block title %}
    顯示現在時間及頁數 {{ offset }}
{% endblock %}

{% block content %}
    {% ifequal offset "99" %}
        <p>這是最後一頁囉！</p>
    {% else %}
        <p>您來到了第 {{ offset }} 頁。</p>
    {% endifequal %}
{% endblock %}
```

如須看網頁的結果，我們還須回到views.py修改樣版的路徑字串，也就是now_template與page_template兩個區域變數。這會跟稍早的結果一樣，請自行嘗試看看，別忘了要在伺服器啟動狀態下連結網頁。

留言板的索引頁 - index.html

我們接下來利用樣版系統做出上一章留言板的網頁瀏覽介面，同樣繼承自base.html，這樣網頁都會出現「現在時間」。總共需要兩個樣版，一個是所有留言的目錄，另一個是個別留言的瀏覽頁。

關於所有留言目錄的樣版index.html，如下。

```
{% extends "template\base.html" %}

{% block title %}
    簡單的留言板
{% endblock %}

{% block content %}
    {% if entry_list %}
        {% for entry in entry_list %}
            <p><a href="{ {{ entry.id }}">{{ entry.title }}</a></p>
        {% endfor %}
    {% else %}
        <p>尚無留言 = _ =</p>
    {% endif %}
{% endblock %}
```

網頁標題設為「簡單的留言板」，然後在{% block content %}裡頭，我們用了{% if %}標籤，檢查是否有留言，若是沒有，網頁顯示「尚無留言」，若有，便利用{% for %}標籤依留言的id屬性，將留言一個標題以一個段落顯示到網頁上。

id屬性繼承自Model物件，由留言的順序從1開始給予整數的序數，這也作為閱覽單獨留言的網址，這個網址「href="{ {{ entry.id }}"」，如第一個留言是指<http://127.0.0.1:8000/1/>的位置。

接著在views.py加入index()函數，如下。

```
def index(request):
    cdt = datetime.datetime.now()
    index_template = r'template/index.html'
    index_context = {'current_time': cdt, 'entry_list': Entry.objects.all()}
    return render_to_response(index_template, index_context)
```

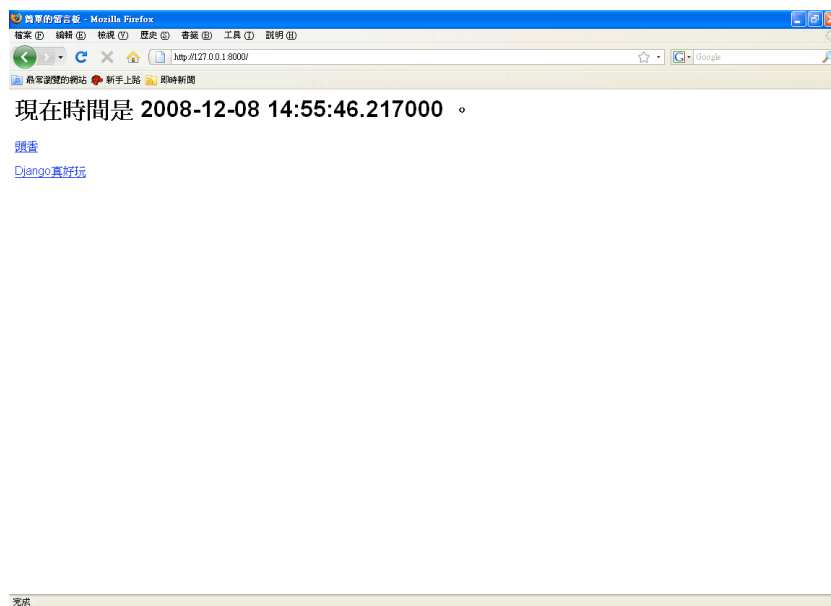
Entry.objects.all()會取得所有已經建立的Entry物件，當然，我們也要先引入Entry物件。

```
from demo.guestbook.models import Entry
```

我們預備把<http://127.0.0.1:8000/>的位置給索引頁，因此urls.py的urlpatterns修改如下。

```
urlpatterns = patterns('',
    (r'^admin/(.*)', admin.site.root),
    (r'^time/$', 'demo.guestbook.views.current_datetime'),
    (r'^time/(\d{1,2})/$', 'demo.guestbook.views.page_counter'),
    (r'^$', 'demo.guestbook.views.index'),
)
```

我們連結到<http://127.0.0.1:8000/>看看吧！



瀏覽個別留言 - entry.html

個別留言的樣版entry.html如下。

```
{% extends "template\base.html" %}

{% block title %}
    {{ entry.title }}
{% endblock %}

{% block content %}
    <h1>留言標題： {{ entry.title }}</h1>
    <h1>留&nbsp;言&nbsp;者： {{ entry.name }}</h1>
    <h1>網&nbsp;址： {{ entry.url }}</h1>
    <p>{{ entry.text }}</p>
    <p><a href="/">回到目錄</a></p>
{% endblock %}
```

網頁標題亦即Entry物件的title屬性，底下依序用<h1>標籤顯示留言標題、留言者、網址，留言本文與的「回到目錄」連結則是用<p>段落標籤，「回到目錄」也就是回到索引頁，所顯示的四個項目分別是Entry物件的四項屬性。

我們在views.py加入entry()函數處理個別留言的網頁顯示。

```
def entry(request, entry_id):
    cdt = datetime.datetime.now()
    message = get_object_or_404(Entry, pk=entry_id)
    entry_template = r'template/entry.html'
    entry_context = {'current_time': cdt, 'entry': message}
    return render_to_response(entry_template, entry_context)
```

這裡用了另一個捷徑函數get_object_or_404()，其能夠取得物件所有的屬性，所以要多引入get_object_or_404的名稱修改如下。

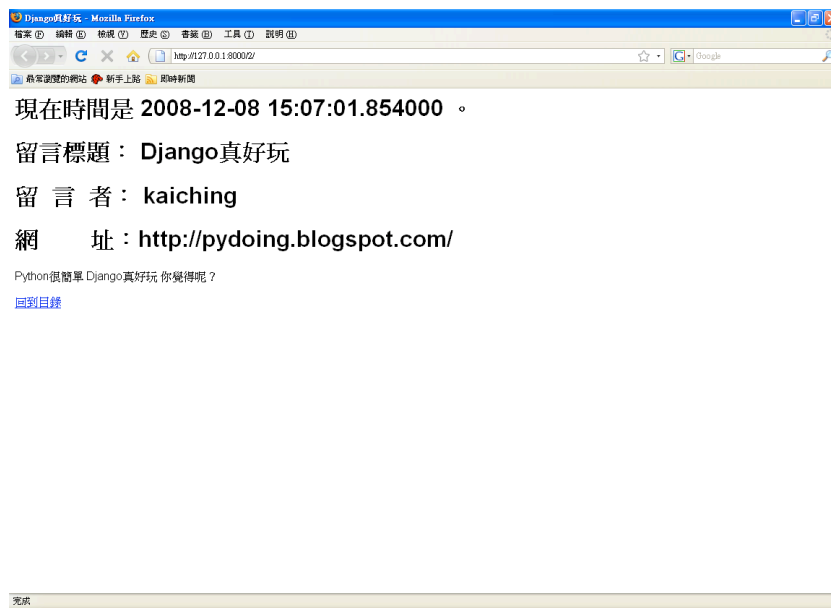
```
from django.shortcuts import render_to_response, get_object_or_404
```

連結網址設定urls.py的urlpatterns加入這一行。

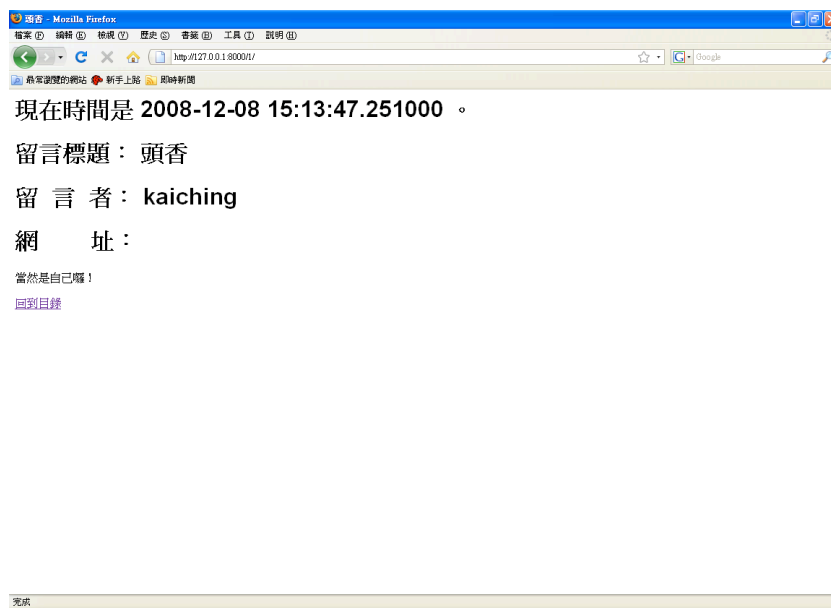
```
(r'^(?P<entry_id>\d+)/$', 'demo.guestbook.views.entry'),
```


注意，「`(?P<entry_id>\d+)`」仍是正規表示法，這可以把連結網址設定為id屬性值。

我們來看看結果吧！先點擊第二個留言「Django真好玩」。



結果如上，我們再點擊「回到目錄」，然後回去看第一個留言。



沒錯，並無出現網址，網頁同時也正常顯示。