

# 網址的對應與委派

1

## 學習目標

### ➡ Django的網址架構

- ➡ URLconf簡介
- ➡ 委派個別的網址到處理函數
- ➡ urlpatterns的Regular Expression語法說明 ( 適用於2.0以前的版本 )

### ➡ 進設定技巧

- ➡ 參數的傳遞
- ➡ include其它整組的urlpatterns設定
- ➡ 為urlpatterns命名

## URLconf簡介

- 是網址的解析以及對應到views.py中函數的主要處理者
- 在Djanog 2.0之前的網址對應主要是以正規表示法 ( Regular Expression ) 來設定網址的內容以及對應的參數格式
- 在Django2.0之後則是以定義路由字串來進行解析

## URLconf處理網址的步驟

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path("", homepage),  
    path('post/<slug>', showpost),  
]
```

```
from minsite.views import homepage, showpost  
  
urlpatterns = [  
    url(r'^$', homepage),  
    url(r'^post/(\w+)$', showpost),  
    #url(r'^post/(.*)$', showpost),  
    url(r'^admin/', admin.site.urls),  
]
```

1. 到`settings.py`中找到`ROOT_URLCONF`的設定，決定要使用哪一個模組。一般來說，大部份的網站都不需要修改這個地方的設定。
2. 載入前述所指定的模組，然後去找到`urlpatterns`這個變數，依據其中的設定來找到對應要處理的網址與函數，它必須是`django.urls.path`或是`django.urls.re_path`（相容於2.0之前的設定方式）的執行實例的Python串列內容。

```
from django.urls import include, path
from django.contrib import admin
from mainsite.views import homepage, showpost
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path("", homepage),
    path('post/<slug:slug>/', showpost),
]
```

## URLconf處理網址的步驟

3. 依照urlpatterns中的順序，一個一個往下核對網址和路由字串中的設定pattern。
4. 發現第一個符合的設定之後，先以HttpRequest的一個實例當做第一個參數，然後再把解析網址中發現的參數依照順序傳遞給後面的處理函數，如果在網址的設定中有對於參數命名的話，則以命名的參數傳遞過去。在這個動作中，基本上就是進行連接到views.py中函數的呼叫程序。
5. 如果都找不到符合的patterns，就會產生一個例外，交由錯誤處理程序。

```
def showpost(request, slug):
    template = get_template('post.html')
    try:
        post = Post.objects.get(slug=slug)
```

## 我們來創建一個新的站點

- `django-admin startproject myUrl`
- `cd myUrl`
- `python manage.py startapp mysite`
- 編輯`urls.py`，先import在`views.py`中定義的模組
- 編輯`urls.py`，建立網址和`views.py`中定義的模組之對應關係
- 執行`python manage.py runserver` 測試網站



## 基本的urls.py內容

```
from django.urls import path
from django.contrib import admin

urlpatterns = [
    path('admin/', admin.site.urls),
]
```

加上首頁homepage的urls.py

```
from django.contrib import admin
```

```
from django.urls import path
```

```
from mysite import views
```

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path("", views.homepage),  
]
```

```
from django.urls import include, path  
from django.contrib import admin  
from mainsite.views import homepage, showpost
```

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path("", homepage),  
    path('post/<slug:slug>', showpost),  
]
```



## 相容於 Django 2.0 以前的 urls.py

```
from django.conf.urls import include, re_path
from django.contrib import admin
from mysite import views
```

```
urlpatterns = [
    re_path(r'^$', views.homepage),
    re_path(r'^admin/', include(admin.site.urls)),
]
```

## 在views.py中的對應程式碼

```
from django.shortcuts import render  
from django.http import HttpResponseRedirect
```

```
def homepage(request):  
    return HttpResponseRedirect("Hello world!")
```



The install worked successfully.

You are seeing this page because you have configured your settings file and you have defined the root URL patterns.

# Hello world!

## 有規則地委派址

- ▶ 假如要委派以下的網址：

[localhost:8000/about/0](http://localhost:8000/about/0)

[localhost:8000/about/1](http://localhost:8000/about/1)

[localhost:8000/about/2](http://localhost:8000/about/2)

[localhost:8000/about/3](http://localhost:8000/about/3)

- ▶ 可以使用如下的pattern完成：

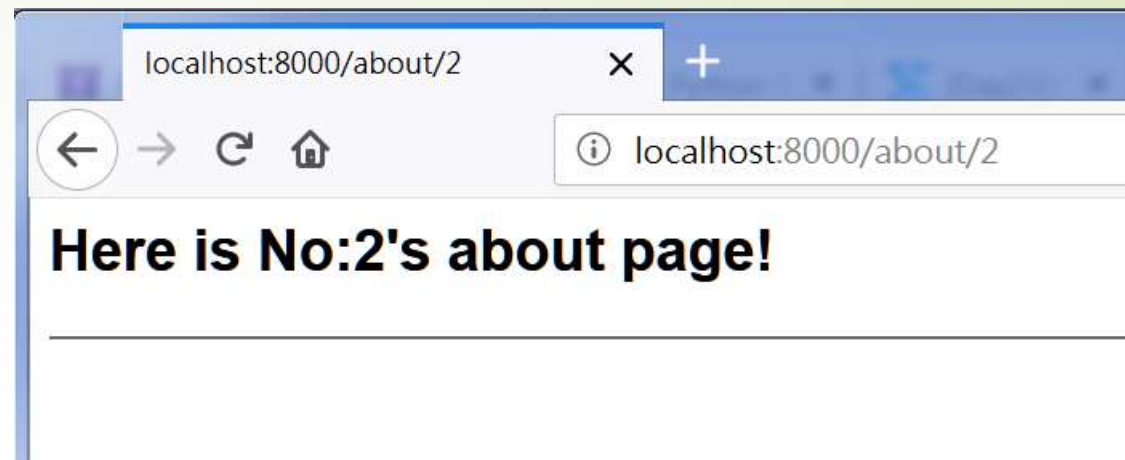
```
path('about/<int:author_no>', views.about),
```

- ▶ `views.py` 對應如下：

```
def about(request, author_no):
```

```
    html = "<h2>Here is No:{}.s about page!</h2><hr>".format(author_no)
```

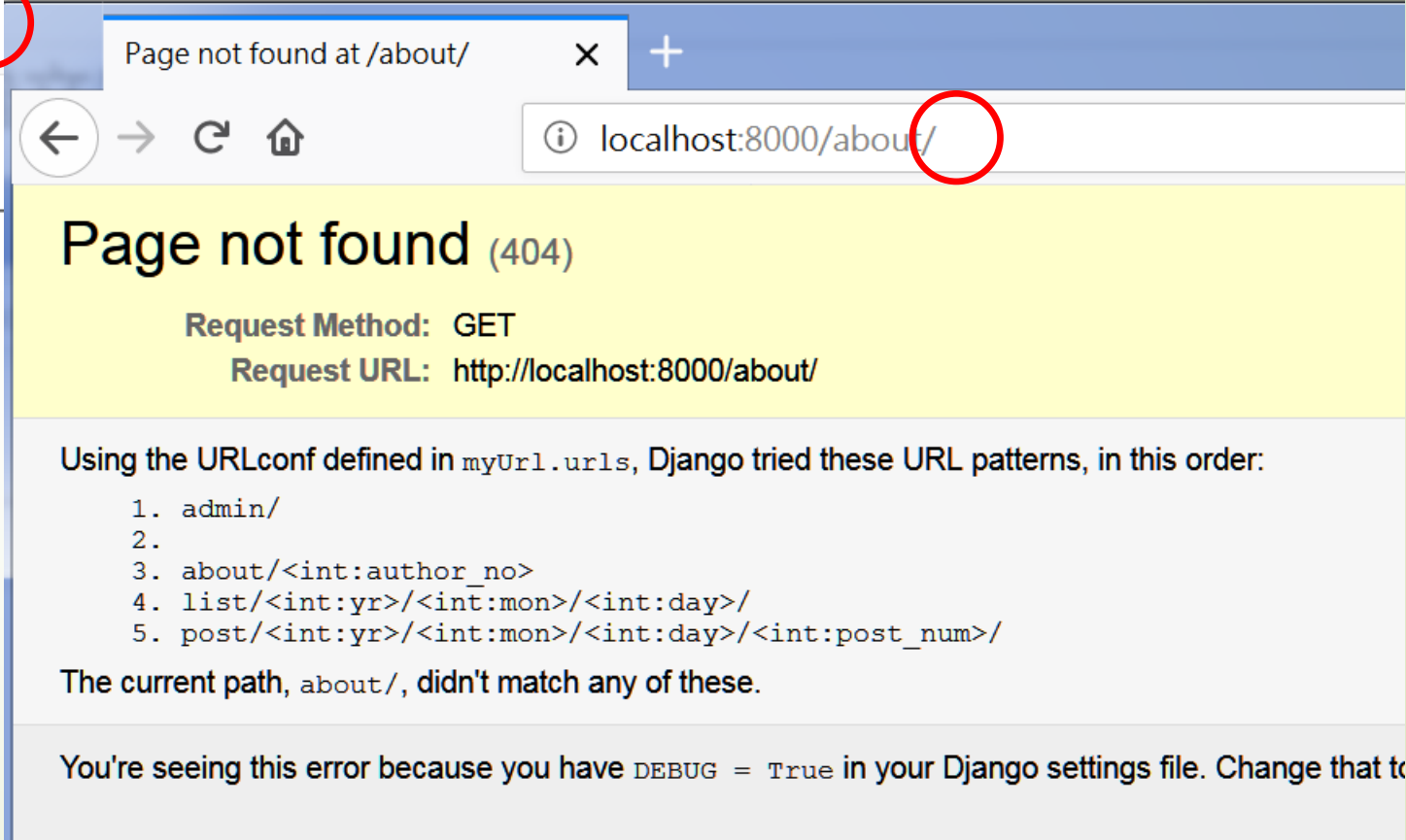
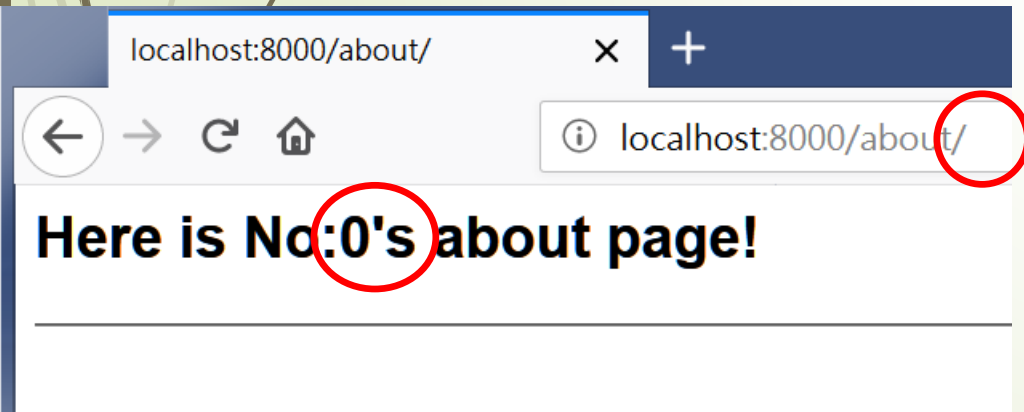
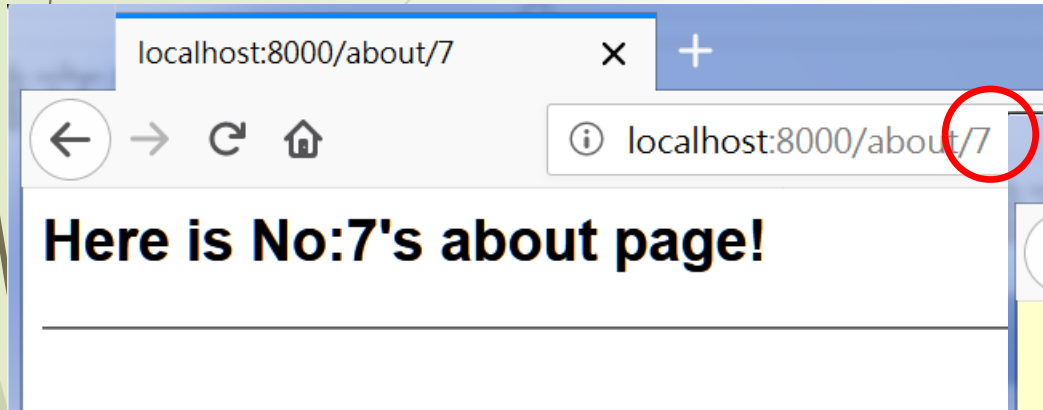
```
    return HttpResponse(html)
```



# Path Converter的參數型態

| 符號   | 說明  |
|------|---|
| str  | 對應參數為字串<br>例：hello  |
| int  | 對應參數為整數<br>例：100  |
| slug | 對應ASCII所組成的字元或符號(像是參數有連字符號或底線等...)<br>例：building-your-1st-django-site |
| uuid | 對應uuid所組的的格式字串。<br>例：075194d3-6885-417e-a8a8-6c931e272f00             |
| path | 對應完整URL路徑，把網址中的"/"視同是參數，而非URL片段，像是上述參數皆為path參數                        |

# 使用預設參數





## url傳遞參數的方法 - 使用預設參數

### ➤ urls.py

```
path('about/', views.about),
```

```
path('about/<int:author_no>/', views.about),
```

### ➤ views.py

```
def about(request, author_no = 0):
```

```
    html = "<h2>Here is Author:{}'s about  
page!</h2><hr>".format(author_no)
```

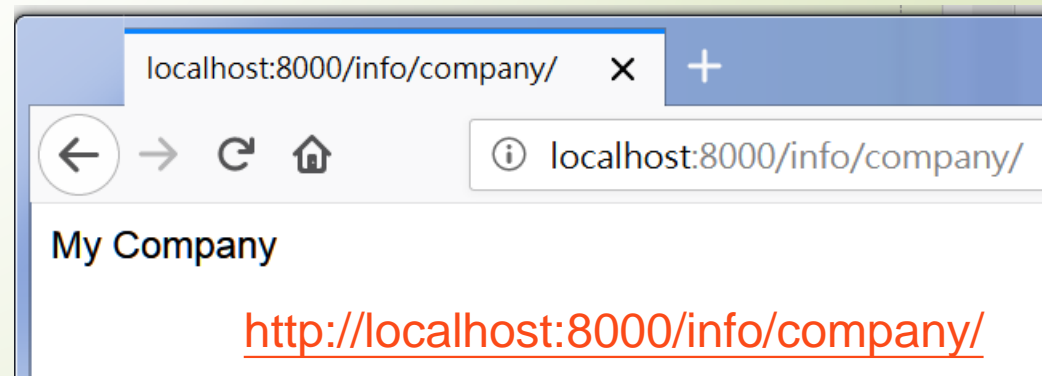
```
    return HttpResponse(html)
```

## include其它urlpatterns的方法

```
from django.conf.urls import include
```

```
my_patterns = [  
    path('company/', views.company),  
    path('sales/', views.sales),  
    path('contact/', views.contact),  
]  
  
urlpatterns = [  
    path('info/', include(my_patterns)),  
]
```

```
def company(request):  
    return HttpResponse("My Company")  
def sales(request):  
    return HttpResponse("My sales")  
def contact(request):  
    return HttpResponse("My contact")
```



# 多個路徑參數

19

<http://localhost:8000/list/2019/10/10>

<http://localhost:8000/post/2019/12/25/07/>

## ► urls.py

```
path('list/<int:yr>/<int:mon>/<int:day>/', views.listing),
```

```
path('post/<int:yr>/<int:mon>/<int:day>/<int:post_num>/', views.post),
```

## ► views.py

```
def listing(request, yr, mon, day):
```

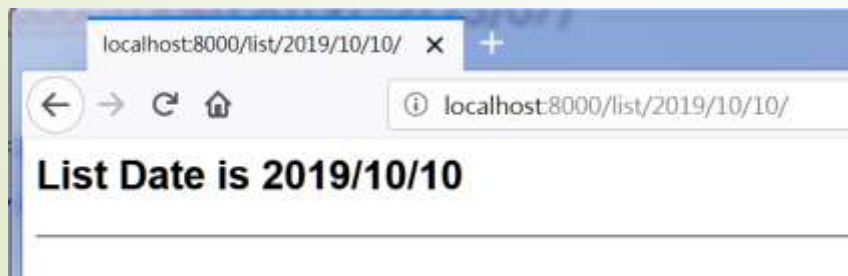
```
    html = "<h2>List Date is {}/{}/{}/</h2><hr>".format(yr, mon, day)
```

```
    return HttpResponse(html)
```

```
def post(request, yr, mon, day, post_num):
```

```
    html = "<h2>{}/{}/{}/Post Number:{}</h2><hr>".format(yr, mon, day, post_num)
```

```
    return HttpResponse(html)
```



## 網址的反解析

➡ 以下寫法很不方便

```
def homepage(request):
```

```
    year = 2019
```

```
    month = 10
```

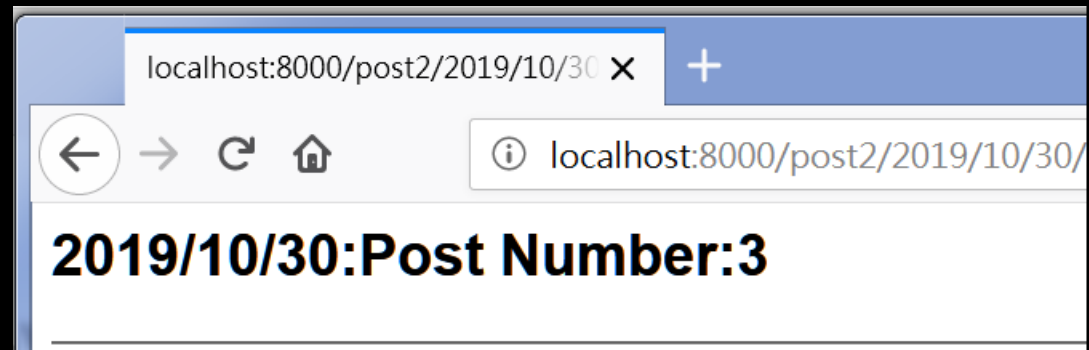
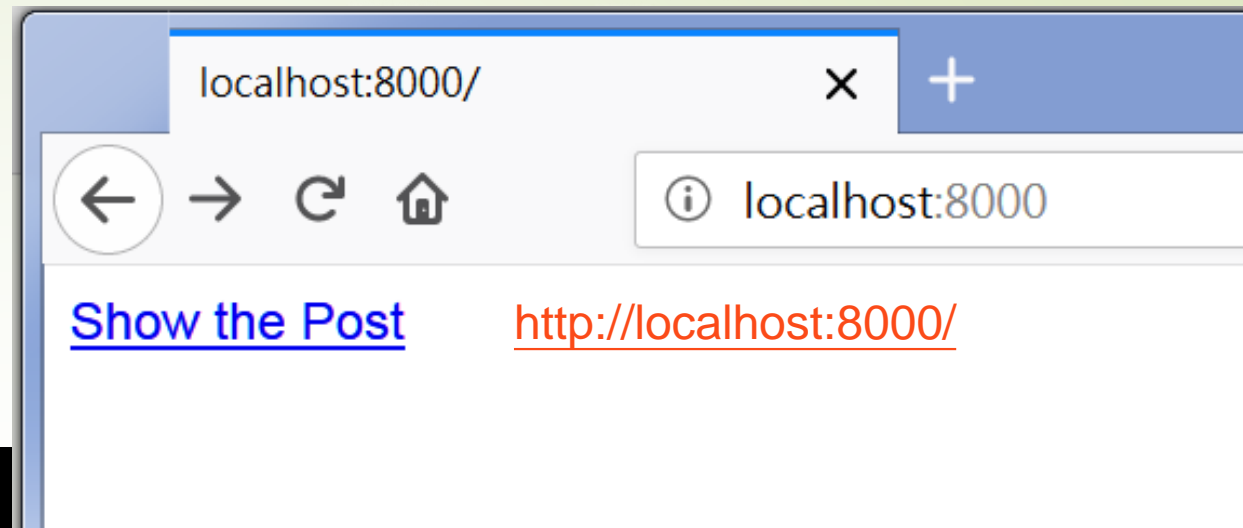
```
    day = 30
```

```
    postid=3
```

```
    html = "<a href='post/{}/{}/{}/{}/'>Show the Post</a>" \
```

```
        .format(year, month, day, postid)
```

```
    return HttpResponse(html)
```



# 使用Python在views.py的寫法

先在urls.py中為網址取個名字：

```
path('post/<int:yr>/<int:mon>/<int:day>/<int:post_num>/', views.post, name='post-url'),
```

```
#from django.core.urlresolvers import reverse
```

```
from django.urls import reverse
```

```
def homepage(request):
```

```
    year = 2019
```

```
    month = 10
```

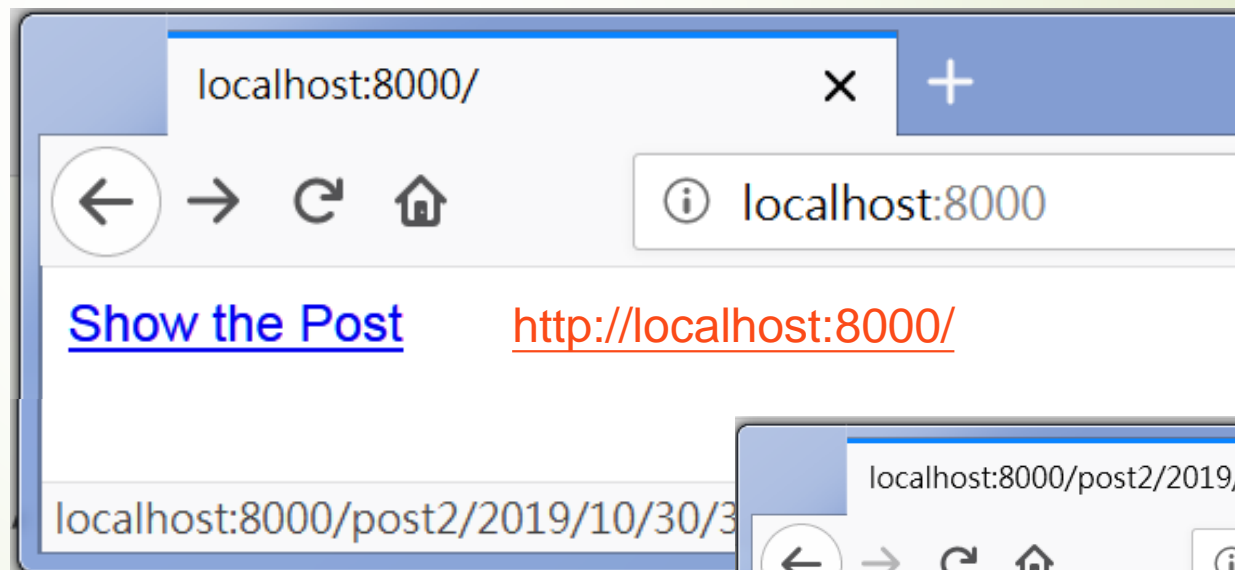
```
    day = 30
```

```
    postid=3
```

```
    html = "<a href='{}'>Show the Post</a>" \
```

```
        .format(reverse('post-url', args=(year, month, day, postid,)))
```

```
    return HttpResponse(html)
```



## 將post加入 template

- 在`settings.py`中設定此資料夾的存取住址
- `'DIRS': [os.path.join(BASE_DIR, 'templates')]`,
- 建立 `templates` 資料夾
- 在`templates`資料夾中建立一個`post2.html`檔案
- 在`views.py`新增
- `def post2(request, yr, mon, day, post_num):`
  - `return render(request, 'post2.html', locals())`



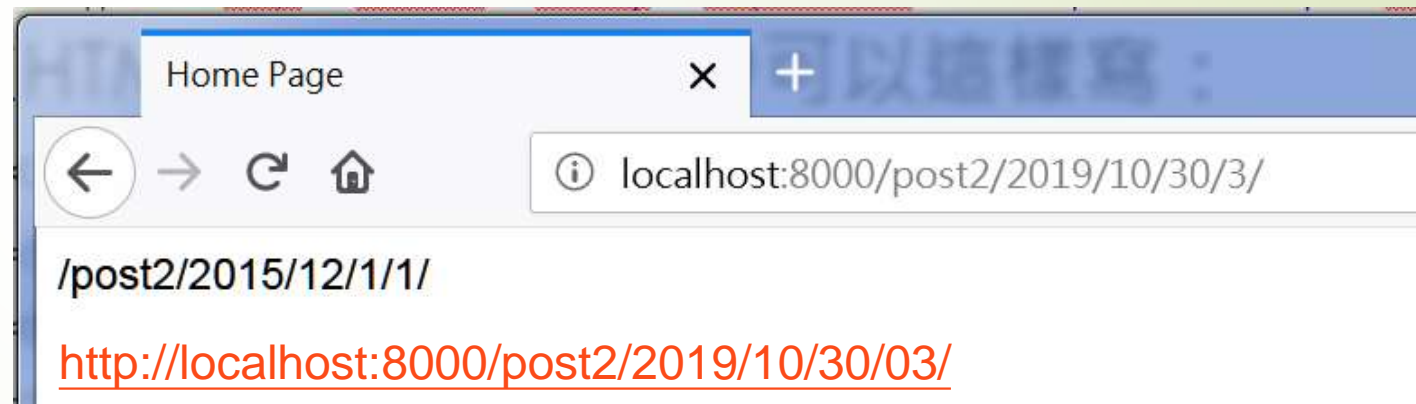
# URLconf反解功能

- 先在urls.py中為網址取個名字：

```
path('post/<int:yr>/<int:mon>/<int:day>/<int:post_num>/', views.post , name='post-url'),  
path('post2/<int:yr>/<int:mon>/<int:day>/<int:post_num>/', views.post2, name='post-url-2'),
```

- 在HTML(post2.html)中可以這樣寫：

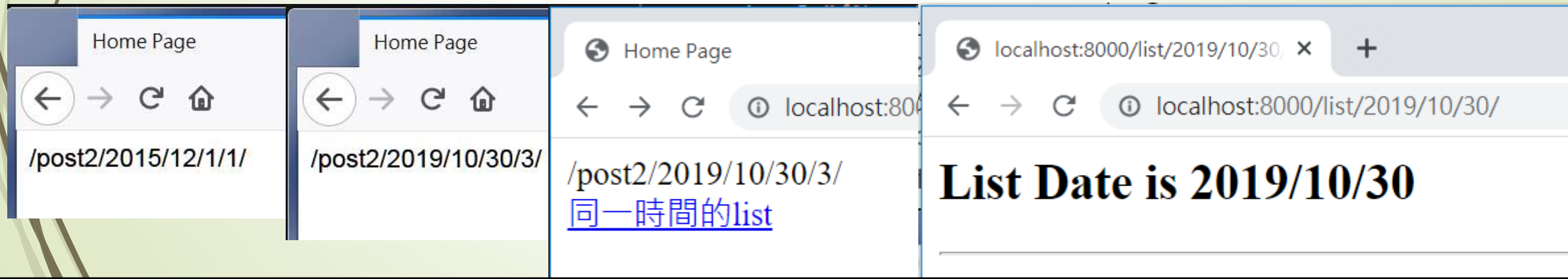
```
<!-- index.html -->  
<!DOCTYPE html>  
<html>  
<head>  
  <meta charset='utf-8'>  
  <title>Home Page</title>  
</head>  
<body>  
  {% url 'post-url-2' 2015 12 1 01 %}  
</body>  
</html>
```



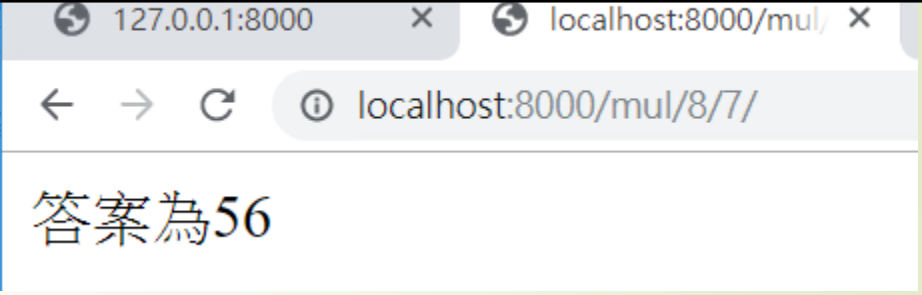
- {% url 'post-url-2' 2015 12 1 01 %}即可解析成/post/2015/12/1/1

## 小練習

- 試修改homepage 的 'post-url' 改成 'post-url-2' 介於post與post2之間看看對homepage來說有什麼變化？
- 請試著將原本網頁中不管網址怎樣都會顯示成「/post2/2015/12/1/1/」修改成依照網址的樣式  
Hint: 修改post2.html的這一行依views.py的變數修改  
{% url 'post-url' 2015 12 1 01 %}
- 請在post2這頁新增一連結，可連到「年月日」相同的list 頁面

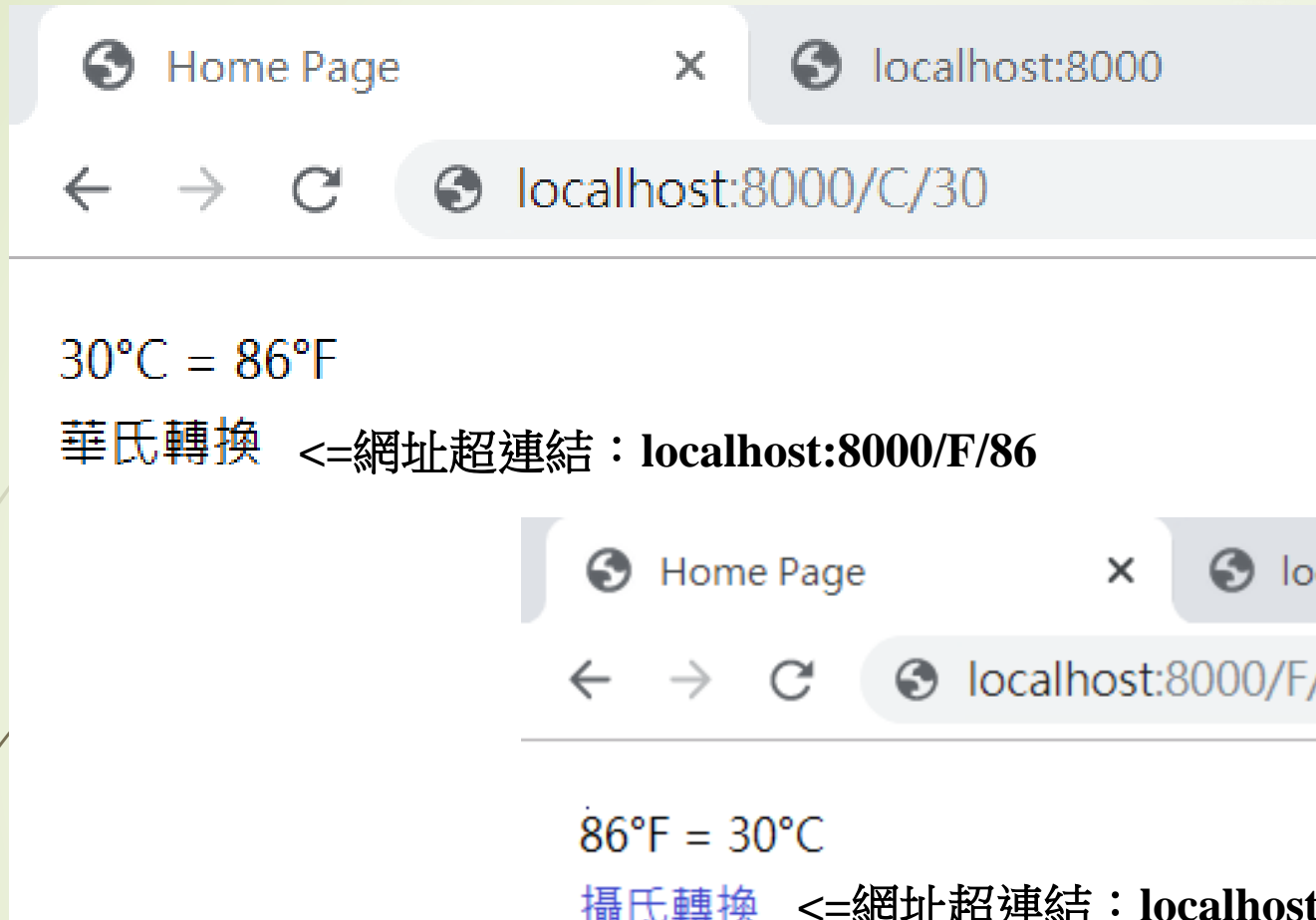


## 回家作業



127.0.0.1:8000 localhost:8000/mul/ x  
localhost:8000/mul/8/7/  
答案為56

1. 請編寫一個簡單的網站，設計一個「/mul/」頁面，可以把「localhost:8000/mul/10/20」後面的2個數字取出，並在網頁中顯示出2數相乘的結果。<http://localhost:8000/mul/8/7/>
2. 請編寫一個簡單的網站，讓使用者可以在網址列中輸入
  1. 「localhost:8000/F/xx」華氏溫度，換算成攝氏溫度顯示在頁面上
  2. 「localhost:8000/C/xx」攝氏溫度，則換算成華氏溫度示在頁面上
    - 攝氏 =  $5/9 * (\text{華氏溫度} - 32)$
    - 華氏 =  $(\text{攝氏})(9/5) + 32$
3. 並在頁面上顯示一連結，內有網址使其可以互相轉換



Home Page x localhost:8000

localhost:8000/C/30

30°C = 86°F

華氏轉換 <=網址超連結：**localhost:8000/F/86**



Home Page x localhost:8000

localhost:8000/F/86

86°F = 30°C

攝氏轉換 <=網址超連結：**localhost:8000/C/30**