

Python 架站特訓班 第二版

Python Website Development Course

django 3

最強實戰

碁峯資訊

【投影片使用規範與聲明】本投影片僅供非營利教學用途，教師得搭配用書授課講解使用，可於用書期間內將投影片置放於學校內部網站，但需有帳密權限機制，且僅供修課學生瀏覽使用。敬請老師善盡著作權保護之責，請勿將投影片任意散布與販售，亦不得以任何形式或方法轉載內容使用。

02

建立專案及 app

2-1 建立 Django 專案

2-2 環境設定

2-3 視圖(view) 與 URL

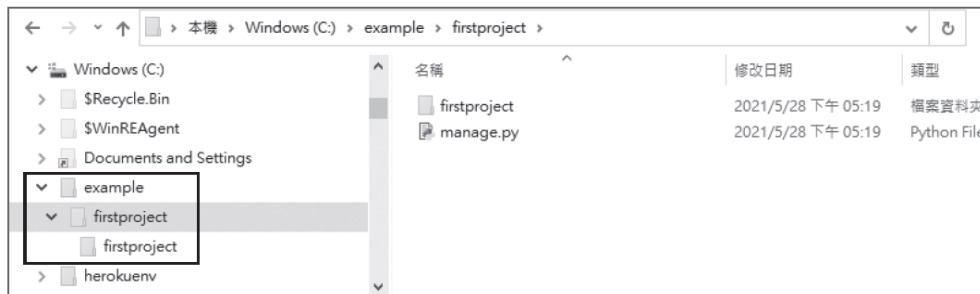
2.1 建立 Django 專案

2.1.1 建立 Django 專案

Django 模組安裝完成後，即可以建立 Django 專案。語法：

```
django-admin startproject 專案名稱
```

專案建立完成會在 <C:\example> 資料夾中建立 <firstproject> 資料夾，同時在 <firstproject> 資料夾下也建立一個 <firstproject> 子資料夾。



firstproject 專案的架構如下：

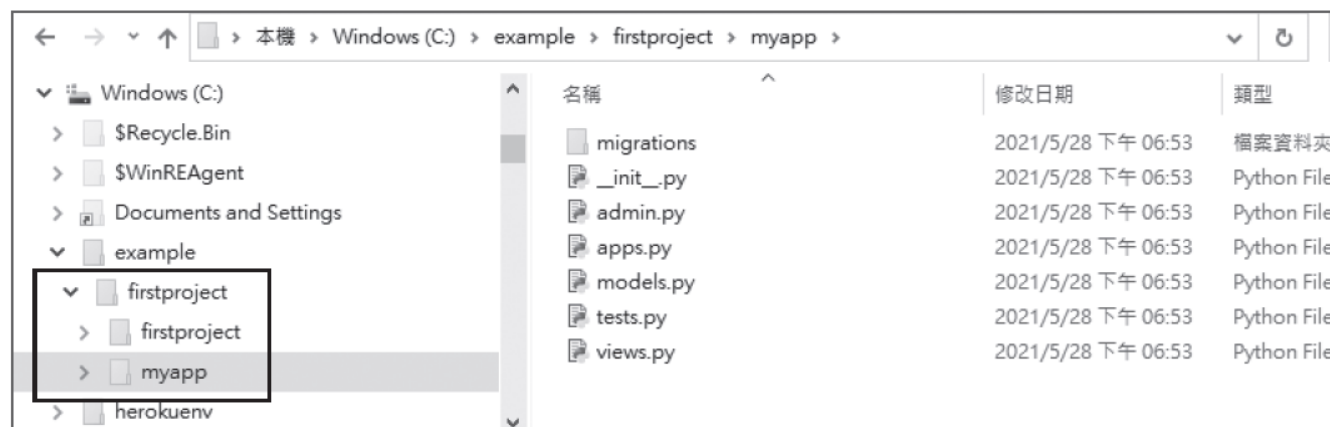
```
\firstproject
├── manege.py
└── \firstproject
    ├── __init__.py
    ├── asgi.py
    ├── settings.py
    ├── urls.py
    └── wsgi.py
```

2.1.2 建立Application 應用程式

建立Application 應用程式的語法：

```
python manage.py startapp 應用程式名稱
```

myapp 應用程式建立完成會在該專案中 (最上層的firstproject) 建立 myapp 資料夾，其檔案架構如下：

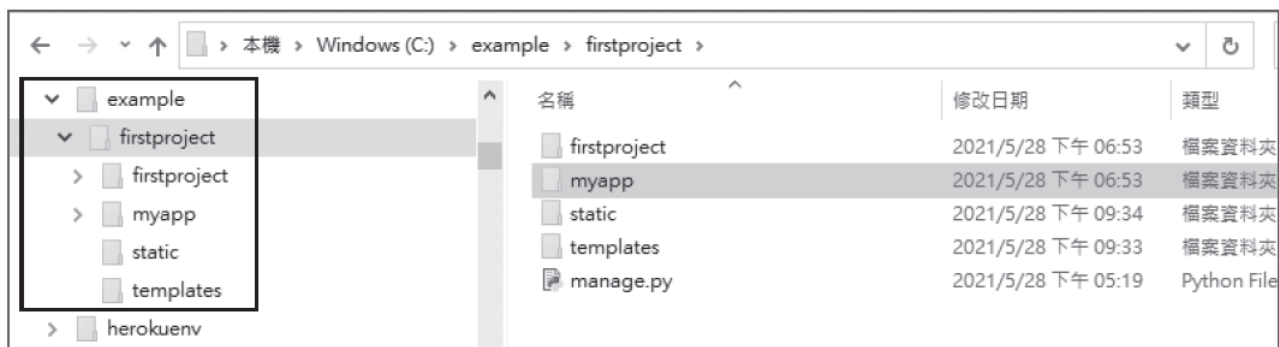


建立 templates 目錄

```
md templates
```

建立 static 目錄

請注意：這兩個目錄是建立在專案的最上層目錄，和myapp 應用程式是平行的，如下圖。



建立 migration 資料檔

```
python manage.py makemigrations
```

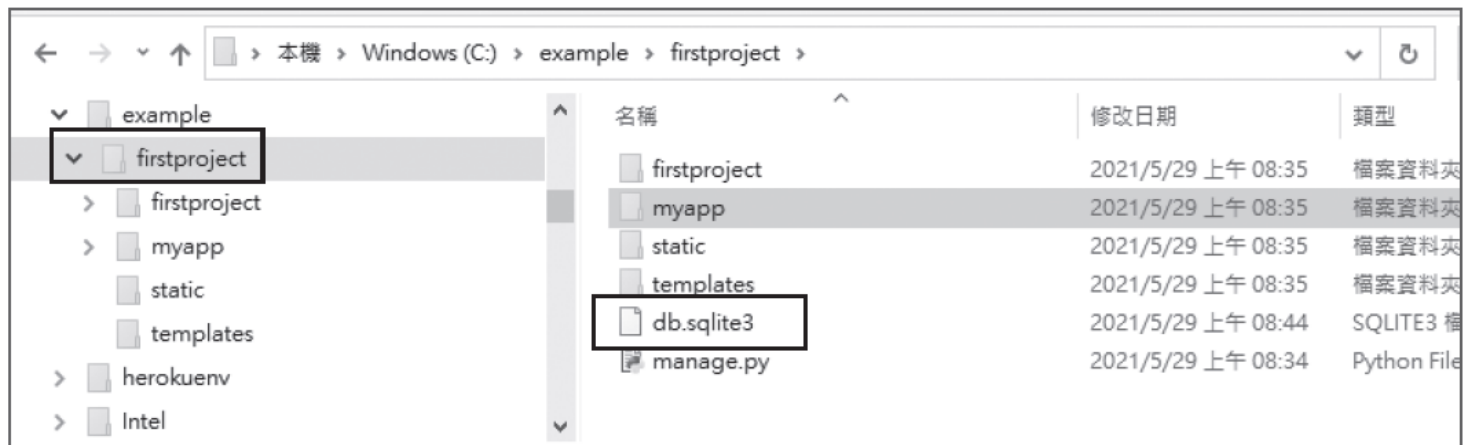
模型與資料庫同步

利用migrate 可以根據migrations 的記錄，將模型同步到資料庫。語法：

```
python manage.py migrate
```

migrate 完成會在專案的最上層目錄下建立<db.sqlite3> 檔，這個檔案就是 Django預設的資料庫檔案。

專案完整建立後的檔案架構如下：

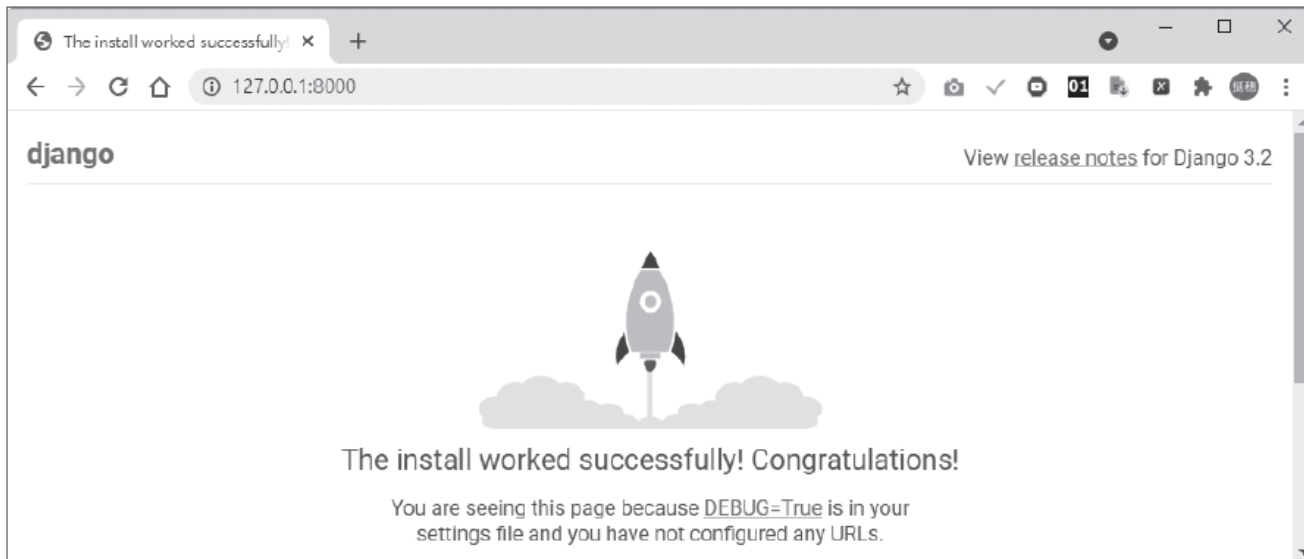


啟動 Server

```
python manage.py runserver
```

瀏覽網頁

開啟瀏覽器輸入「<http://127.0.0.1:8000>」即可以看到本地端的網頁。



2-2 環境設定

請打開<settings.py> 檔案，如下完成設定。

除錯模式設定

```
26     DEBUG = True
```

加入 Application 應用程式

```
33     INSTALLED_APPS = [  
34         'django.contrib.admin',  
35         'django.contrib.auth',  
36         'django.contrib.contenttypes',  
37         'django.contrib.sessions',  
38         'django.contrib.messages',  
39         'django.contrib.staticfiles',  
40         'myapp', # 新增的 app  
41     ]
```


設定 Template 路徑

```
55  TEMPLATES = [  
56      {  
57          'BACKEND': 'django.template.backends.django.DjangoTemplates',  
58          'DIRS': [BASE_DIR / 'templates'], # 加上 templates 路徑  
59          'APP_DIRS': True,  
.....
```

設定語系和時區

預設語系為英文語系，請更改為中文語系，台北時區。

```
107  LANGUAGE_CODE = 'zh-hant' # 改為繁體中文  
  
109  TIME_ZONE = 'Asia/Taipei' # 改為台北時區
```

設定 static 靜態檔的路徑

```
121  STATIC_URL = '/static/'  
122  STATICFILES_DIRS = [ # 加入 static 路徑  
123      BASE_DIR / 'static',  
124  ]
```

2-3 視圖(view) 與 URL

Django 的程式架構是採用 urlpattern 網址和函式對照方式，主要有兩個步驟：

1. 設定<urls.py> 檔urlpatterns 串列中 url 網址和函式的對照。
2. 在<views.py> 中撰寫函式。

2.3.1 設定 urls.py

```
firstproject\urls.py
```

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

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

url 語法

urlpatterns 串列中url 設定網址和函式的對照，語法如下：

```
path( 網址 , 函式 ),
```

網址是指加在「127.0.0.1:8000/」後面的網址，通常由網址名稱及參數組合而成：

```
網址名稱 /< 資料型態一 : 參數一 >/< 資料型態二 : 參數二 >/.....
```

```
firstproject\urls.py
```

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

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

```
from myapp.views import sayhello
```

```
urlpatterns = [
```

```
    path('admin/', admin.site.urls),
```

```
    path('', sayhello),
```

```
]
```

2.3.2 定義函式

```
myapp\views.py
```

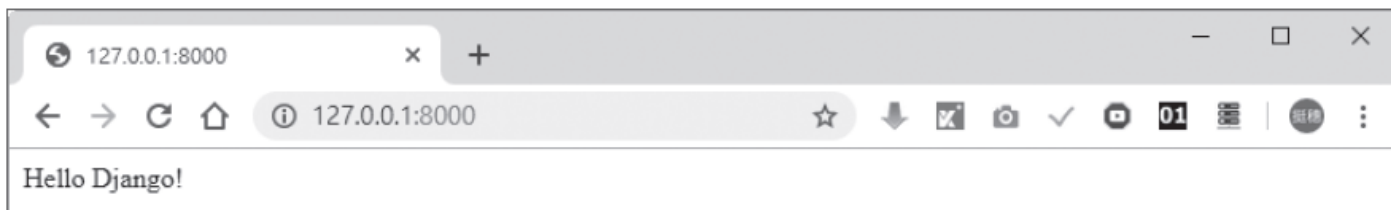
```
.....略
```

```
from django.http import HttpResponse
```

```
def sayhello(request):
```

```
    return HttpResponse("Hello Django!")
```

完成之後，瀏覽「127.0.0.1:8000/」這個網頁，可看到如下的畫面。



網址可以直接傳送參數，例如：瀏覽「127.0.0.1:8000/hello2/ 李小明」這個網址，會顯示「Hello 李小明」，也就是要傳送參數「李小明」。

```
firstproject\urls.py
```

```
.....略
```

```
from myapp.views import sayhello, hello2
```

```
urlpatterns = [
```

```
    .....略
```

```
    path('hello2/<str:username>', hello2),
```

```
]
```

最後再定義hello2 自訂程序，並以username 接收參數。

```
myapp\views.py  
.....略  
from django.http import HttpResponse  
.....略  
def hello2(request, username):  
    return HttpResponse("Hello " + username)
```

執行結果：



2.3.3 模版的使用

```
templates\hello3.html
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <meta charset='utf-8'>
```

```
    <title> 第一個模版 </title>
```

```
</head>
```

```
<body>
```

```
    <h1> 歡迎光臨：{{username}} </h1>
```

```
    <h2> 現在時刻：{{now}} </h2>
```

```
</body>
```

```
</html>
```

模版中讀取變數的語法為：

```
{{ 變數 }}
```

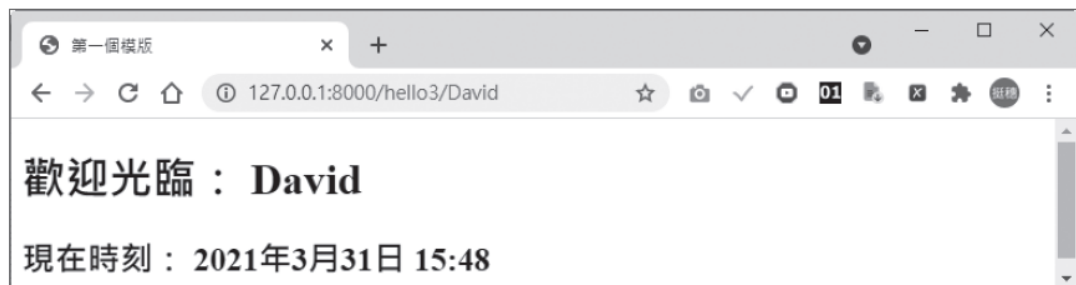
定義<urls.py> 中瀏覽「hello3/xxx」執行hello3 自訂程序，如下：

```
firstproject\urls.py  
from myapp.views import sayhello,sayhello2,hello3  
urlpatterns = [  
    .....略  
    path('hello3/<str:username>', hello3),  
]
```

同時將參數傳遞給<hello3.html> 模版。

```
myapp\views.py  
from datetime import datetime  
def hello3(request,username):  
    now=datetime.now()  
    return render(request,"hello3.html",locals())
```

完成後執行「127.0.0.1:8000/hello3/David」，顯示結果如下：



2.3.4 加入 static 靜態檔案

在 <settings.py> 設定檔的 122~124 列 STATICFILES_DIRS 已設定專案下的 static 為靜態檔案路徑。

```
121  STATIC_URL = '/static/'  
122  STATICFILES_DIRS = [      # 加入 static 路徑  
123      BASE_DIR / 'static',  
124  ]
```

但是光宣告還不夠，要在網頁上顯示圖片或設定 css 樣式靜態檔案，還必須在.html檔中以「{% load static %}」宣告使用靜態檔案，同時以「{% static 靜態檔案 %}」格式設定靜態檔案的路徑，以實例說明如下：

請在<static> 資料夾新增一個<images> 資料夾，並將<ball.png> 圖片檔放在<images> 資料夾中，同時新增一個<css> 資料夾，然後將<style.css> 檔放在<css> 資料夾中。

定義 url，瀏覽 hello4/xxx 執行 hello4 自訂程序，如下：

firstproject\urls.py

```
from myapp.views import sayhello,hello2,hello3,hello4

urlpatterns = [
    .....略
    path('hello4/<str:username>', hello4),
]
```

myapp\views.py

```
def hello4(request,username):
    now=datetime.now()
    return render(request,"hello4.html",locals())
```

templates\hello4.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <title> 顯示圖片的模版 </title>
    {% load static %}
    <link href="{% static "css/style.css" %}" rel="stylesheet"
        type="text/css" />
</head>
<body>
    <div id="home">
        
        <span class="info"> 歡迎光臨：{{username}}</span>
        <h2> 現在時刻：{{now}} </h2>
    </div>
</body>
</html>
```

<style.css> 定義 info 類別樣式，如下：

```
static/css/style.css  
.info {  
    color:red;  
    font-size: 1.5em;  
}
```

完成後執行「127.0.0.1:8000/hello4/David」，顯示結果如下：

