

Network Programming

- 網路程式設計 -

發送電子郵件與購物車

授課教師：張珀銀 老師



前一單元內容回顧

11

專題：購物車

11-1 發送電子郵件

11-2 購物車

1

發送電子郵件

11.1 發送電子郵件

使用線上郵件程式是一個非常實用的技巧，使用者可以很方便的寄送電子郵件。python 已內建Smtplib 模組，不必另行下載安裝，只要匯入該模組即可利用gmail 帳號寄送電子郵件。

11.1.1 設定 Google 帳號權限

要利用gmail 帳號寄送電子郵件，除了必須要有Google 帳號之外，還必須調整Google 帳號的權限：啟用 **安全性較低的應用程式存取權限**。

1. 開啟 Google 網站後登入Google 帳號，若無帳號請先申請。
2. 選按右上角帳號的圖示，再點選 **管理你的Google 帳戶** 進入Google 帳戶頁面。點選 **安全性**，將螢幕向下捲動，到 **低安全性應用程式存取權** 處，點選 **開啟存取權(不建議)**。



11.1.2 寄送電子郵件基本語法

利用gmail 帳號寄送電子郵件首先要匯入smtpplib 模組，設定主機及寄送者gmail 帳號及密碼：

```
import smtpplib  
主機變數 = "smtp.gmail.com:587"  
帳號變數 = "gmail 帳號 "  
密碼變數 = "gmail 密碼 "
```

接著設定郵件內容、標題及收件者：

```
內容變數 = " 郵件內容 "  
訊息變數 = MIMEText( 內容變數 )  
訊息變數 ["Subject"] = " 郵件標題 "  
收件者變數 = " 收件者電子郵件 "
```

最後是連接寄信主機將信件寄出：

```
伺服器變數 = SMTP( 主機變數 )    # 建立 SMTP 連線
伺服器變數 .ehlo()    # 跟主機溝通
伺服器變數 .starttls()    #TLS 安全認證
伺服器變數 .login( 帳號變數 , 密碼變數 )    # 登入
伺服器變數 .sendmail( 帳號變數 , 收件者變數 , 訊息變數 .as_string() )    # 寄信
伺服器變數 .quit()    # 關閉連線
```

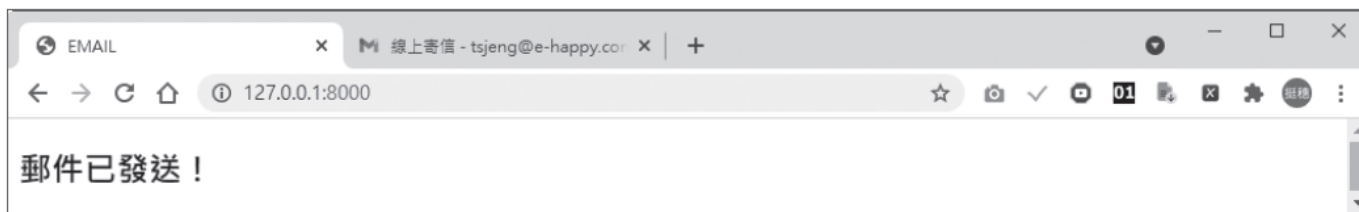
例如：

```
server = SMTP(strSmtplib)    # 建立 SMTP 連線
server.ehlo()    # 跟主機溝通
server.starttls()    #TLS 安全認證
server.login(strAccount, strPassword)    # 登入
server.sendmail(strAccount, mailto, msg.as_string())    # 寄信
server.quit()    # 關閉連線
```


寄送電子郵件範例

將本章範例sendmail 專案複製到複製到<C:\example> 資料夾中，然後切換到sendmail 專案以「python manage.py runserver」啟動伺服器。

首頁 (127.0.0.1:8000) 會顯示「郵件已發送！」訊息。



到收件者信箱可見到已成功收到信件。



11.1.3 寄送電子郵件進階設定

寄送 html 格式電子郵件

前一小節使用 MIMEText 僅傳送郵件內容一個參數，MIMEText 的語法為：

```
MIMEText( 郵件內容 [, 郵件格式 , 郵件編碼 ] )
```

郵件格式：常用的值為「plain (純文字)」及「html (網頁格式)」，預設值為「plain」。

郵件編碼：通常使用「utf-8」，如此可以顯示各種文字，預設值即為「utf-8」。

同時寄送多人電子郵件

收件者變數的資料型態也可使用串列，執行結果相同：

```
mailto = [" 收件者電子郵件 "]
```

若串列中包含多個電子郵件元素，則系統會傳送信件給每一個電子郵件信箱：

```
mailto = [" 收件者電子郵件一 ", " 收件者電子郵件二 ", .....]
```

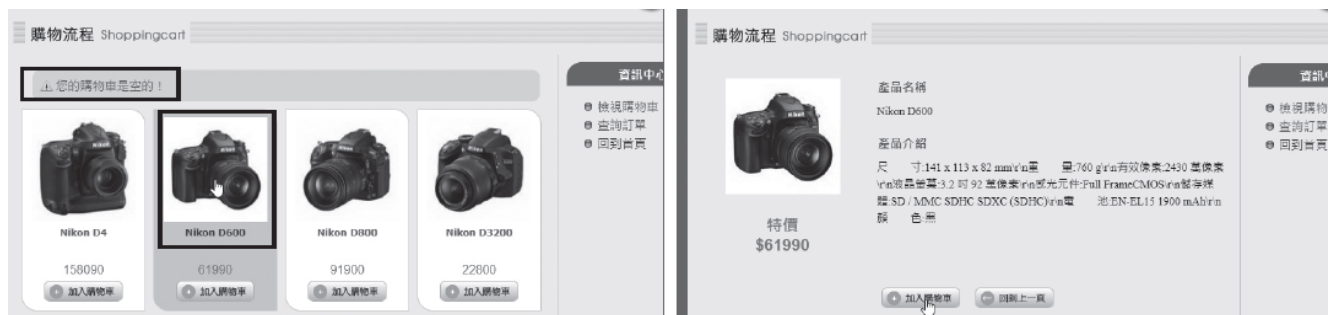
11-2 購物車

想要在網站上自己開店當老闆嗎？那麼購物車就是您網站必須具備的主要功能之一，本專題可以幫助您完成購物流程的規劃、製作及網路開店的夢想！

11.2.1 使用購物車

將本章範例cart 專案複製到複製到<C:\example> 資料夾中，修改<views.py> 中第109-110 列為使用者的 Gmail 帳號及密碼，然後在命令提示字元視窗切換到 cart 專案，以「python manage.py runserver」啟動伺服器。

首頁會顯示所有商品照片、名稱及價格，此時尚未開始購物，會顯示「購物車是空的」訊息。點選商品照片或名稱會切換到該商品詳細頁面。在首頁或商品詳細頁面按 **加入購物車** 鈕可將商品加入購物車並切換到購物車頁面。

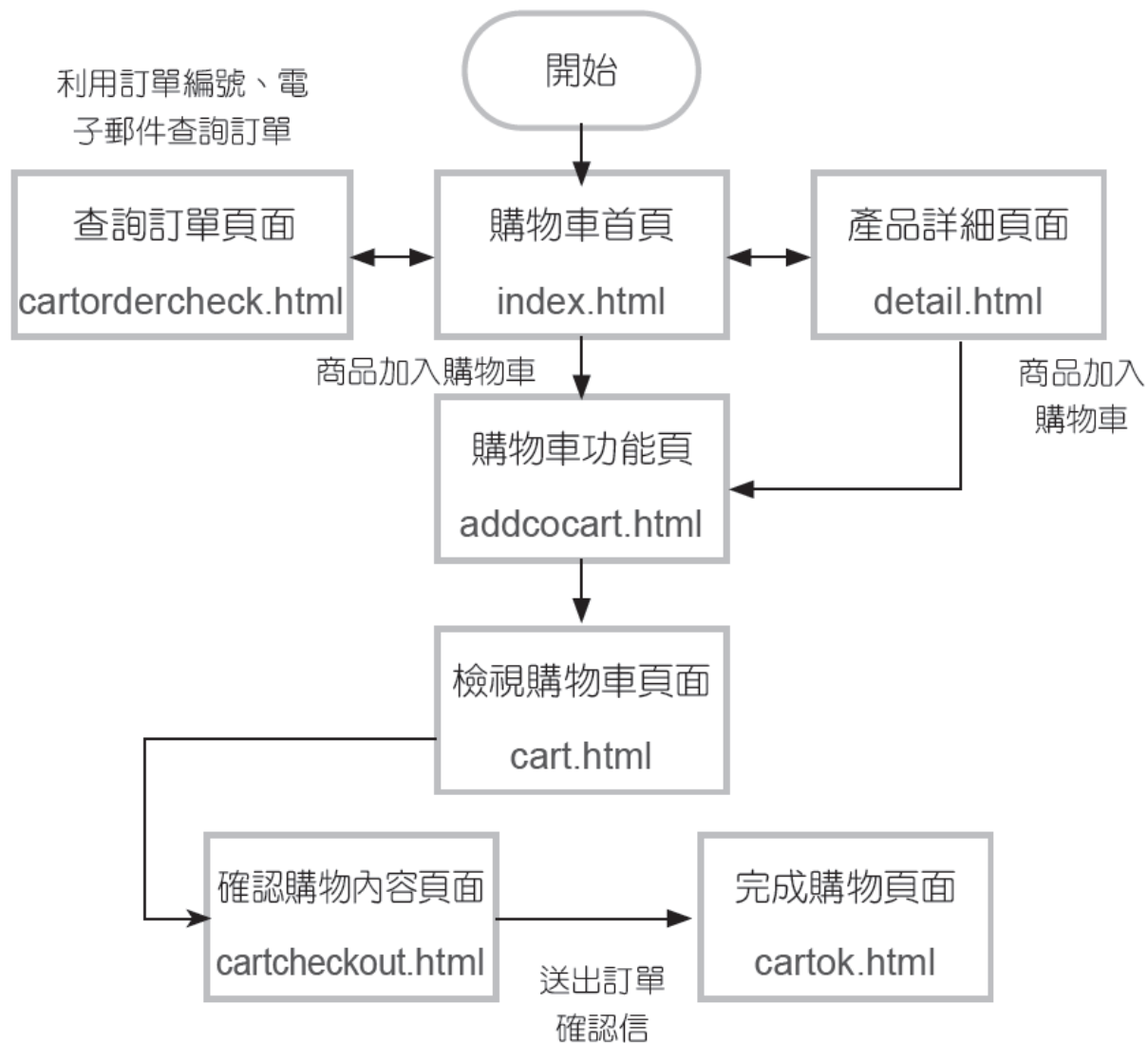




2

購物車

11.2.2 購物車流程圖



11.2.3 購物車資料庫結構

資料庫結構定義在<models.py> 中， 內含ProductModel、OrdersModel 及 DetailModel 資料表：

```
cartapp\models.py
```

```
1 from django.db import models
2
3 class ProductModel(models.Model):
4     pname = models.CharField(max_length=100, default='')
5     pprice = models.IntegerField(default=0)
6     pimages = models.CharField(max_length=100, default='')
7     pdescription = models.TextField(blank=True, default='')
8     def __str__(self):
9         return self.pname
10
```

```
11 class OrdersModel(models.Model):
12     subtotal = models.IntegerField(default=0)
13     shipping = models.IntegerField(default=0)
14     grandtotal = models.IntegerField(default=0)
15     customname = models.CharField(max_length=100, default='')
16     customemail = models.CharField(max_length=100, default='')
17     customaddress = models.CharField(max_length=100, default='')
18     customphone = models.CharField(max_length=100, default='')
19     paytype = models.CharField(max_length=50, default='')
20     def __str__(self):
21         return self.customname
22
23 class DetailModel(models.Model):
24     dorder = models.ForeignKey('OrdersModel', on_delete=models.CASCADE)
25     pname = models.CharField(max_length=100, default='')
26     unitprice = models.IntegerField(default=0)
27     quantity = models.IntegerField(default=0)
28     dtotal = models.IntegerField(default=0)
29     def __str__(self):
30         return self.pname
```

11.2.4 Url 配置檔

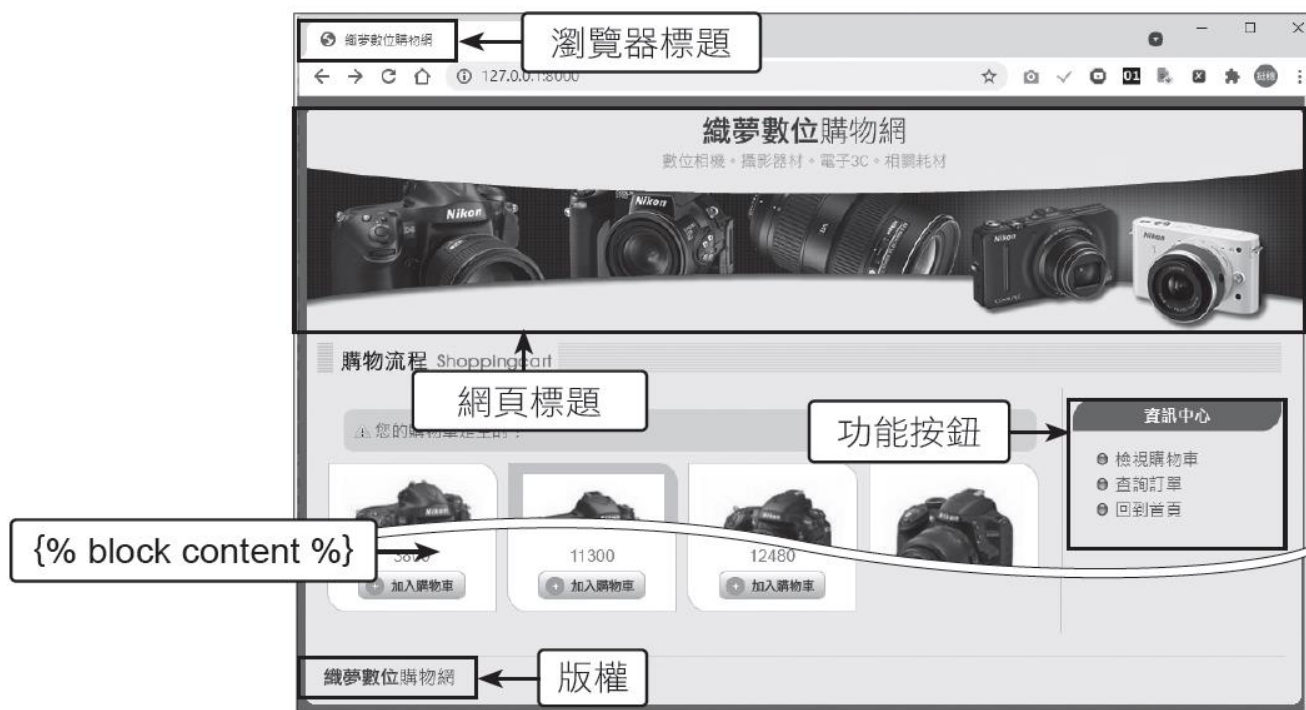
cart\urls.py

略.....

```
16 from django.contrib import admin
17 from django.urls import path
18 from cartapp import views
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', views.index),
23     path('index/', views.index),
24     path('detail/<int:productid>', views.detail),
25     path('addtocart/<str:ctype>', views.addtocart),
26     path('addtocart/<str:ctype>/<int:productid>', views.addtocart),
27     path('cart/', views.cart),
28     path('cartorder/', views.cartorder),
29     path('cartok/', views.cartok),
30     path('cartordercheck/', views.cartordercheck),
31 ]
```


11.2.5 建立網頁基礎模版

網站中所有網頁結構類似，為這些網頁建立基礎模版檔<base.html>：包括瀏覽器標題、網頁標題、右方**資訊中心**功能按鈕及下方的版權部分。



11.2.6 首頁處理函式及模版

使用者以「`http://127.0.0.1:8000`」開啟本專題就會執行<views.py> 中的index 函式載入<index.html> 網頁，會顯示所有商品。

11.2.7 商品詳細頁面處理函式及模版

使用者在首頁按圖片或商品名稱就會執行<views.py> 中的detail 函式載入<detail.html> 網頁顯示商品的詳細資訊。

11.2.8 顯示購物車頁面處理函式及模版

使用者在任何頁面按右方 **檢視購物車** 或在首頁、商品詳細頁面按 **加入購物車** 鈕就會執行<views.py> 中的cart 函式載入<cart.html> 網頁，顯示購買商品的詳細資訊並可修改購買數量。

11.2.9 購物車處理函式

購物車處理函式(addtocart) 負責購物車中商品的增添、修改及刪除等功能，本函式沒有顯示頁面，處理完相關功能後會切換到相關頁面。

首先是使用者在首頁或商品詳細頁面按 **加入購物車** 鈕的處理程式，功能是將使用者選取的商品加入購物車。

11.2.10 訂單頁面處理函式及模版

使用者在顯示購物車頁面點選 **我要結帳** 鈕就會執行<views.py> 中的cartorder 函式載入<cartorder.html> 網頁讓使用者填寫購買者資訊。

11.2.11 訂購完成頁面處理函式及模版

使用者在訂單頁面按 **確認購買** 鈕就會執行<views.py> 中的cartok 函式載入<cartok.html> 網頁。注意：第109-110 列需改為使用者的Gmail 帳號及密碼。

11.2.12 查詢訂單頁面處理函式及模版

使用者在任何頁面點選右方 **查詢訂單** 鈕就會執行<views.py> 中的cartordercheck函式載入<cartordercheck.html> 網頁。



Final-term

1. HTML:

依循網址教學完成範例 [URL](#)

2. FLASK:

依循網址教學完成範例 [URL](#)，建置於
PythonAnywhere