

# INFOR Academy

INFOR 30th 蔣立元 周柏融 李宗奇 周柏宇

## 1. 介紹

INFOR Academy 是一個網站，統整了30th幹部們一年來的教學資料，以及學習上曾遇到的問題，提供循序、條列式的內容，以及清晰的邏輯概念，讓自學程式語言既方便又有效率，也可以輔助社課教學，突破程式語言很難的刻板印象，讓社團整體學術力獲得提升。

## 2. 製作動機

我們每天晚上都有開放放學社課。雖然剛開學時來參加的人少說也有10位以上，但隨著時間過去，人數越來越少，歷屆的學長們也有遇到此種情形。學弟們剛開始應該也都對資訊有些興趣，那是什麼原因導致人數越來越少呢？我們發現許多學弟上完我們的社課之後，回家可能想要深入學習或者是練習一下，但在網路上找不到連續完整的內容，又捨不得買書。或者是想要複習社課內容，但沒有講師講解的社課講義很難讀懂。因此我們想製作一個網站，統整教學資料，讓學弟不必再網路上盲目地找資料，可以輕鬆自學程式語言。

## 3. 開發工具、方法介紹

### 3.1 Virtualenv

Python 環境使用一段時間後都會安裝許多套件，Virtualenv是個能創造虛擬且獨立的Python開發環境，使得各個專案彼此互不干擾，避免架設網站時浪費許多寶貴的網站資源。

首先先安裝virtualenv。

```
$ pip install virtualenv
```

接著到你想要放檔案的資料夾內，使用virtualenv指令建立虛擬環境。filename改成虛擬環境的名稱。--python參數可以設定你要的python版本。接著啟動虛擬環境，此時你的command line前面會有(filename)。使用deactivate指令即可結束虛擬環境。

```
$ virtualenv filename --python=python3
$ source filename/bin/activate
(filename) $ deactivate
```

### 3.2 Django

Python最為人稱道的特性便是可以透過安裝許多套件不斷擴充其功能，當然肯定少不了網站架構的套件。Django是目前最多人使用的Python網站開發框架，開發快速、效能高，具備資料庫元件，強大的錯誤訊息顯示，致力於淺顯易懂和優雅的程式碼等等。因此有許多著名網站都是使用此套件來進行開發。



### 3.2.1 建立專案

Django套件安裝好後，透過指令建立專案，projectname填入專案名稱。

```
$ django-admin startproject projectname
```

此時會產生多個檔案，介紹常用的一些：

manage.py	Python命令檔，可以用它執行多種操作，包含建立APP、同步資料庫、啟動server等等。
settings.py	本專案的設定檔。
urls.py	url的配置檔。
wsgi.py	網頁伺服器和Django的介面設定檔。

建立應用程式，建立templates資料夾放顯示的模板，建立static資料夾放靜態檔案。並在settings.py檔案中加入新增的App、設定templates、static資料夾路徑，並設定時區、語系。基礎的配置大致上設定好之後，便可使用指令啟動伺服器。

```
$ python manage.py runserver
```

### 3.2.2 Django的MTV架構

Django架構採用Model、Template和View，Model負責處理資料庫的資料，Template負責處理頁面顯示，View則是控制與整合資料庫和顯示之間的互動。

### 3.2.3 View與Template的互動

首先先定義View。打開建立好的app資料夾，開啟views.py，輸入以下code。並在templates資料夾內加入index.html,hello.html。

```
from django.shortcuts import render

def index(request):
    return render(request, "index.html")

def hello(request,name):
    message = "Hello " + name
    return render(request, "hello.html", locals())
```

接著設定URL。開啟urls.py，輸入以下code。app改成你的App名稱。

```
from django.urls import path,include
from app import views

urlpatterns = [
    path('', views.index),
    path('hello/<str:name>', views.hello),
]
```

由以上的程式碼來看，path()定義了當你瀏覽某個網址時要執行什麼程序。當你使用瀏覽器瀏覽https://localhost:8000/時render()呼叫index.html，因此會顯示index.html的內容。在path()中用<>來捕捉url上的值，以下為常見格式。

str	除了/ 以外的非空字串	int	正整數包含0
slug	字母、數字橫槓下滑線組成的字串	path	任何非空字串，包含/

locals()則是能將所以區域變數轉換成字典傳遞到template。在template則可以使用兩個大括號來顯示變數{{}}。使用瀏覽器瀏覽https://localhost:8000/hello/Pooh時，網頁就會顯示：Hello Pooh。

```
<!-- hello.html -->
<!DOCTYPE html>
<html>
<head>
    <title>hello</title>
</head>
<body>
    <p>{{ message }}</p>
</body>
</html>
```

Template可以透過表單(form)傳送資料給網頁後端，action屬性指定傳送的目標頁面，method屬性則是傳送的方法，常用的為GET和POST。以下為一個簡單的小範例。以POST傳送資料必須加入{% csrf\_token %}這一行啟動CSRF防護。

```
<!-- index.html -->
<!DOCTYPE html>
<html>
<head>
    <title>hello</title>
</head>
<body>
    {% if message == "no" %}
        <form action = "/index" method = "POST" name = "form">
            {% csrf_token %}
            <input type="text" name="name">
            <input type="submit" value="submit">
        </form>
    {% else %}
        <p>{{ message }}</p>
    {% endif %}
</body>
</html>
```

```
from django.shortcuts import render

def index(request):
    if request.method == "POST":
        message = request.POST['name']
    else message = "no"
    return render(request, "index.html", locals())
```

若要在Views中使網頁重定向的話，可以使用以下方法，參數可放完整的url或者是絕對路徑。

```
from django.http import HttpResponseRedirect
```

```
def gotoindex(request):  
    return HttpResponseRedirect('/index')
```

### 3.2.4 Template 語言

已知取用變數要使用兩個大括號`{{}}`，但使用語法和Python並不相同，以下事常用對照表。在Template中兩個大括號就像標籤，只要有if或for敘述，最後就要加上`{{endif}}`或`{{endfor}}`，若for中的list沒有任何資料便會執行`{{empty}}`的區塊。當需要對變數進行操作時，需要使用過濾器(filter)，預設的filter很多，大家可以在官網找到很多資料，當然也可以自定義filter。

類別	Template 語法	Python 語法
字典	<code>{{ dic.name }}</code>	<code>dic[name]</code>
屬性	<code>{{ obj.name }}</code>	<code>obj.name</code>
函式	<code>{{ obj.name }}</code>	<code>obj.name()</code>
串列	<code>{{ list.0 }}</code>	<code>list[0]</code>
條件敘述	<code>{{ if condition }}</code> <code>{{ elif condition }}</code> <code>{{ else }}</code> <code>{{ endif }}</code>	<code>if condition:</code> <code>elif condition:</code> <code>else:</code>
迴圈	<code>{{ for i in list }}</code> <code>{{ empty }}</code> <code>{{ endfor }}</code>	<code>for i in list:</code>
對變數進行操作	<code>{{ value add:"2" }}</code>	<code>value = value+2</code>

### 3.2.5 建立資料庫

Django原生支援許多資料庫，內建預設的為sqlite，但只要經過簡單的設定可以輕鬆從sqlite轉換到其他資料庫。首先先在model.py中定義資料類別，完成後，輸入以下指令，將建立資料表的架構和版本記錄下來，並且同步到資料庫。

```
$ python manage.py makemigrations  
$ python manage.py migrate
```

以下程式碼建立兩個資料型態，必須繼承`models.Model`，再分別建立欄位。第一個為課程，第二個為文章。`CharField`為單行輸入的字串資料，`DateTimeField`為日期時間格式，`TextField`為多行輸入的字串資料，`PositiveIntegerField`為正整數資料。可以在小括號內加入參數，例如`auto_now`可以使其自動更新為當前時間。更多`models.Model`的欄位資料可以在官網上找到。系統也會自動產生一個欄位叫做`id`，由0開始遞增。然而有時候資料型別之間也有連結關係，課程可能會包含很多文章，但每篇文章都只屬於一個課程，因此是多對一(Many-to-one)的關係，`ForeignKey`便能達到此種連結，`related_name`設定了反查時的關鍵字，`on_delete`設定了當連結到的資料被刪除的話該執行什麼操作。

```

from django.db import models

class Course(models.Model):
    Topic = models.CharField(max_length = 50)
    def __str__(self):
        return self.Topic

class Article(models.Model):
    Title = models.CharField(max_length = 50, unique = True)
    Author = models.CharField(max_length = 20, blank = True)
    Time = models.DateTimeField(auto_now = True)
    Content = models.TextField(max_length = 10000)
    Quantity = PositiveIntegerField(default = 0)
    Course = models.ForeignKey(Course, related_name = "Article",
                               on_delete=models.PROTECT)

    def __str__(self):
        return (self.Title)

```

建好資料型態之後，就可以在view.py import 該資料型態，使用以下程式碼讀取資料庫的資料，也可以直接取得全部資料的串列。以下使用Course當範例。

```

Course.objects.get(Topic = "Python")
Course.objects.all()

```

新增資料和刪除資料則可以用以下程式碼。

```

#create
unit = Course.objects.create(Topic = "Java")
unit.save()
#delete
unit.Course.objects.get(id = 1)
unit.delete()

```

### 3.2.6 後台管理

Django已有現成的後台管理系統。在urls.py加入以下程式碼。

```

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

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

```

然後在App底下的admin.py檔案中將你的資料型別import進去並register。classname為資料型別的名稱。

```

from django.contrib import admin
from app import models

admin.site.register(models.classname)

```

接著利用以下指令註冊admin帳號。

```
$ python manage.py createsuperuser
```

你便可以透過<https://localhost:8000/admin>來登入後台管理系統。

### 3.3 MathJax



MathJax是一個Javascript的函式庫，可以在瀏覽器上透過css生成數學符號。只需要引入MathJax的檔案，便可以使用MathML或Latex兩種數學表示語法在網頁中顯示複雜的數學公式。

### 3.4 highlight.js



highlight.js是一個Javascript的函式庫，可以在網頁上讓程式碼高亮顯示。這類工具很多，會選擇highlight.js是因為他支援的語言很多、配色方案也多，使用上非常容易，只要引入highlight.js便可以使用以下標籤在網頁上高亮顯示程式碼。理論上會自動偵測程式語言，當然也可以加class屬性指定程式語言。

```
<pre>
  <code class="python" >
    <!--content-->
  </code>
</pre>
```

### 3.5 網站前端

前端的部分，將使用Bootstrap、W3School進行排版、美化。

## 4. 未來展望

目前只打算撰寫一些課程的教學內容，例如Python基礎語法、C++基礎語法、演算法和資料結構等等，期待未來能新增更多元的內容，也能增加線上即時評測學習成果的功能。

## 5. 參考資料

- 5.1 <http://www.books.com.tw/products/0010762818>
- 5.2 <https://djangogirlstaipei.gitbooks.io/django-girls-taipei-tutorial/>
- 5.3 <https://www.djangoproject.com/>
- 5.4 <https://www.mathjax.org/>
- 5.5 <https://highlightjs.org/>
- 5.6 <https://getbootstrap.com/>
- 5.7 <https://www.w3schools.com/>
- 5.8 <https://github.com/twtrubiks/django-field-tutorial>