

Web應用程式開發 Web Application Development

CSS & JSON

授課教師：林器弘

1

Outline

- CSS 妝點註冊網頁
- JSON 操作

2

上課前準備

- 下載上課檔案week6.zip。

3

註冊網頁結構1

```
week6/  
├── app.py  
├── templates/  
│   ├── register.html  
│   └── result.html
```

4

註冊網頁畫面

• 註冊頁

127.0.0.1:9000/register

註冊

姓名：

學號：

Email：

系所：

連絡電話：

建議：

成功頁面

註冊成功！

姓名：123

學號：123

Email：123@123.123.tw

單位：人工智慧學系

連絡電話：1231231231

建議：123

[回到註冊頁](#)

5

註冊網頁完成結構2

week6 /

```
├── app.py
├── static/
│   ├── style.css
│   └── send.png
└── templates/
    ├── register.html
    └── result.html
```

6

套用CSS讓版面變好看

註冊頁面



成功頁面



7

CSS - body

```
/* 設定網頁body */
Body {
  font-family: Arial, "Microsoft JhengHei";
  background-color: #f3f3f3;
}

/* 標題 */
h1 {
  text-align: center;
  color: #333;
}
```

8

自定義容器

- 可以透過以下 HTML 範例，將某一個範圍進行框選，並定義成某一個具有圖像、標題和一些文字的卡片組合。

styles.css

```
.container {  
  max-width: 500px; /* 最大寬度 */  
  margin: auto; /* 水平置中 */  
  background: #fff; /* 白色背景 */  
  border: 1px solid #ddd; /* 1px 灰色邊框 */  
}
```

CSS1.html

```
<div class="container">  
  <h2>Card title</h2>  
  <p>Card content</p>  
</div>
```

9

9

建立註冊頁 container (1/2)

- Width: 400 px; /*設定寬度像素*/
- Margin : 80px auto; /*外邊距，外部留白*/
 上下留 80px 空白；左右自動置中
- background-color : white; /*設定背景顏色*/
- border-radius: 10px; /*區塊的邊角變成圓角*/
- text-align: left; /*文字對齊*/

```
margin-top: 80px;  
margin-right: auto;  
margin-bottom: 80px;  
margin-left: auto;
```

10

建立註冊頁 container (2/2)

/*區塊加上陰影*/

- box-shadow: 0 0 10px rgba(0,0,0,0.1);
/* 0 = 水平位 (向右偏移 0px)
0 = 垂直位移 (向下偏移 0px)
10px = 模糊半徑 (陰影擴散程度)
rgba(0,0,0,0.1) = 顏色(RGB三原色) (黑色透明度 10%) */

/*內容與邊框之間的距離*/

- padding 是內邊距:20px 30px;
/*上下 20px ;
左右 30px 。*/

11

建立結果頁 container

```
.container2 {  
  background-color: white;  
  width: 400px;  
  margin: 80px auto;  
  padding: 30px;  
  border-radius: 10px;  
  box-shadow: 0 0 10px rgba(0,0,0,0.1);  
  text-align: left;  
}
```

12

CSS - button

```
/* 設定按鈕 */
button {
    background: none;
    border: none;
    cursor : wait;
/*改變游標形狀*/
}

button img {
    width: 40px;
    height: 40px;
}

button:hover {
    background-color: #0056b3;
}

.button-right{
    text-align: right;
/* 讓裡面的內容靠右 */
}
```

13

CSS - 輸入元件

```
/* 輸入元件 */
input, select, textarea {
    width: 100%;
    padding: 8px;
    border-radius: 5px;
    border: 1px solid #ccc;
    box-sizing: border-box;
    font-size: 14px;
}
```

14

CSS – 連結跟段落設定

```
form p {                               /*依所在位置指定，表單(form)裡面的段落(p)*/
    margin: 15px;
}
/* 返回連結 */
a {
    display: inline-block;
    margin-top: 15px;
    color: #007bff;
    text-decoration: none;
}
a:hover {
    text-decoration: underline;
}
```

15

url_for

- url_for(endpoint, **values)
- endpoint：就是「路由函式名稱（即 @app.route 下方的函式名）」或是 "static"（指靜態檔案）。
- values：若路由有參數，可以在這裡傳入。

16

url_for

```
@app.route("/")
def index():
    return "這是首頁"
```

```
@app.route("/register")
def register():
    return "註冊頁"
```

```
from flask import redirect, url_for
```

```
return redirect(url_for("register"))
```

- `url_for("register")` 會自動產生 `"/register"`。
即使你未來把路由改成 `/signin`，這裡也不需要改程式，因為 Flask 會根據函式名稱重新產生正確的 URL。

17

HTML 模板（Jinja2）中使用

- Flask 會把 `url_for()` 傳入模板引擎，所以你可以直接在 HTML 這樣寫：

```
<a href="{{ url_for('register') }}">返回註冊頁</a>
```

- 同樣，它會自動變成：

```
<a href="/register">返回註冊頁</a>
```

18

靜態檔案（例如 CSS、圖片、JS）

- Flask 會自動幫你設定靜態/static資源資料夾。
你可以這樣引用：

```
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
    >> /static/style.css

    >> /static/logo.png
<script src="{{ url_for('static', filename='main.js') }}"></script>
    >> /static/main.js
```

19

有參數的路由

- 若你的路由有參數，例如：

```
@app.route("/user/<name>")
def user(name):
    return f'Hello, {name}!'
```

- 那你可以用：

```
url_for("user", name="Alice")
```

- Flask 結果為：

```
/user/Alice
```

20

在按鈕中放入圖片

```
<button type="submit" title="送出表單"> 送出表單</button>
```



```
<button type="submit" title="送出表單">  
      
</button>
```

21

加入CSS

```
<head>  
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">  
</head>  
  
<body>  
    <div class="container1">  
        .....  
    </form>  
    </div>  
</body>
```

22

Register.html

```
<p>
  連絡電話：<input type="tel" name="phone" pattern="[0-9]{10}"
Placeholder="輸入十碼的手機號碼" required>
</p>

<div class="button-right">
  <button type="submit" title="送出表單" >
    
  </button>
</div>
```

23

限制輸入長度

- HTML 中的 minlength 屬性定義了使用者可以在 <input> 或 <textarea> 元素中輸入的最小字串長度。
- 此屬性用於表單驗證，確保提交的值符合指定的最小長度要求。
- minlength 的主要特性：適用元素：它可與 <input> 元素（如：type="text"、type="email"、type="password"、type="search"、type="tel"、type="url" 的元素）和 <textarea> 元素一起使用。
- 值：此屬性的值必須是非負整數（0 或更大），表示所需的最小 UTF-16 代碼單元數。
- 反之maxlength 亦然






24

支援瀏覽器版本

- maxlength

				
Yes	Yes	Yes	Yes	Yes

- minlength

				
40.0	17.0	51.0	10.1	27.0

25

練習一

- 完成設計自己的註冊網頁。
- 字形改為標楷體與Times New Roman
- 當中在學號中的位置使用Placeholder提示出你自己的學號。
- 設計當電話號碼超過十碼時，不能輸入(不能出現在畫面上)

26

JSON

27

JSON

- JavaScript Object Notation (JSON) 可將結構化資料 (structured data) 呈現為 JavaScript 物件的標準格式，常用於網站上的資料呈現、傳輸 (例如將資料從伺服器送至用戶端，以利顯示網頁)。
- 本文將說明Python 搭配 JSON 時所應知道的觀念，包含如何在 JSON 物件中存取資料項目，並寫出你自己的 JSON。
- JSON是一種流行的數據格式，用於表示結構化數據。在服務器和 Web 應用程序之間以 JSON 格式傳輸和接收數據是很常見的。
- 我們將藉由一些範例學習在 Python 中解析、讀取和編寫 JSON的檔案格式。

28

JSON

- 雖然 JSON 是以 JavaScript 語法為基礎，但可獨立使用，且許多程式設計環境亦可讀取 (剖析) 並產生 JSON。
- JSON 可能是物件或字串。當你想從 JSON 中讀取資料時，JSON 可作為物件；當要跨網路傳送 JSON 時，就會是字串。
- JSON 物件可儲存於其自有的檔案中，基本上就是副檔名為 .json 的文字檔案，以及 application/json 的 MIME type。

29

JSON

- JSON 是一種輕量化的資料交換格式（尤其在網頁與 API）
- 語法類似於 Python 的字典與清單
- 結構：物件 {}、陣列 []、字串、數字、布林值、null
- 用途：API 資料傳遞、設定檔、Web 應用資料交換、日誌/事件。
- 幾乎所有語言都支援（Python、JavaScript、C#、Go...）

30

JSON

- 一個JSON檔案的結構:

```
{
  "name": "Alice",
  "age": 20,
  "skills": [ "Python", "HTML"],
  "is_student": Yes
}

{
  "languages": "English",
  "name": "Bob",
  "numbers": [
    2,
    1.6,
    null
  ]
}
```

31

JSON

```
{
  "squadName" : "Super hero squad",
  "homeTown" : "Metro City",
  "formed" : 2016,
  "secretBase" : "Super tower",
  "active" : true,
  "members" : [
    {
      "name" : "Molecule Man",
      "age" : 29,
      "secretIdentity" : "Dan Jukes",
      "powers" : [
        "Radiation resistance",
        "Turning tiny",
        "Radiation blast"
      ]
    },
    {
      "name" : "Madame Uppercut",
      "age" : 39,
      "secretIdentity" : "Jane Wilson",
      "powers" : [
        "Million tonne punch",
        "Damage resistance",
        "Superhuman reflexes"
      ]
    },
    {
      "name" : "Eternal Flame",
      "age" : 1000000,
      "secretIdentity" : "Unknown",
      "powers" : [
        "Immortality",
        "Heat Immunity",
        "Inferno",
        "Teleportation",
        "Interdimensional travel"
      ]
    }
  ]
}
```

32

JSON

- 我們在上面提過「JSON 物件基本上就是 JavaScript 物件，而這敘述在大多數情況下都對」。
- 上述「在大多數情況下都對」的理由，就是因為陣列也可以是有有效的 JSON 物件，例如：

```
[
  {
    "name" : "Molecule Man",
    "age" : 29,
    "secretIdentity" : "Dan Jukes",
    "powers" : [
      "Radiation resistance",
      "Turning tiny",
      "Radiation blast"
    ]
  },
  {
    "name" : "Madame Uppercut",
    "age" : 39,
    "secretIdentity" : "Jane Wilson",
    "powers" : [
      "Million tonne punch",
      "Damage resistance",
      "Superhuman reflexes"
    ]
  }
]
```

33

JSON

- In Python,

```
p = '{"name": "Bob", "age": "20", "sex": "F", "ID": 221,
    "languages": ["Python", "HTML"]}'
```

- 同時，我們也可以把上面的字串轉換成JSON object 的檔案
- 要使用 JSON（字符串或包含 JSON object 的文件），您可以使用 Python 中的 json module。您需要先導入 module，然後才能使用它。

```
import json
```

34

JSON

- 主要函式：dumps()、loads()、dump()、load()
- dumps / loads：在記憶體內轉換
 - dumps()：將Python資料轉換為JSON格式
 - loads()：將JSON格式資料轉換為Python資料。
- dump / load：與檔案操作結合
 - dump()：將Python資料轉換為JSON檔案，資料將轉換為JSON格式。
 - load()：從JSON檔案中讀取資料，資料將轉換為Python格式。

35

JSON 資料轉為 Python 內部的資料格式

原始 JSON 類型	轉為 Python 的類型
object	dict
array	list
string	str
number (int)	int
number (real)	float
true	True
false	False
null	None

36

Python dict to JSON

- 如果你有一組python的dict資料集。
- 我們會使用json.dumps() method，將這組python資料集轉成json格式。
- 將Python資料結構轉換為JSON(dumps()方法返回一個str)：

```
import json

person_dict1 = {'name': 'Bob', 'age': 20, 'sex': "F " }
#輸出為字串
person_json = json.dumps(person_dict1)
print(person_json)
# Output: {"name": "Bob", "age": 20, "sex": "F"}
```

37

Python dict to file

- 如果處理的是檔案而不是字串，可以使用 json.dump() 和 json.load() 來編碼和解碼JSON資料。
- 我們使用 'w' 以書寫模式打開了一個名為 firstjson.Json 的文件。如果該文件尚不存在，則會創建該文件。
- 例如：

```
data = json.loads(person_json)

# 寫入檔案
with open('firstjson.json', 'w') as f:
    json.dump(data, f)
```

38

Writing JSON to a file

- 印出來確定的json 內容後，確定資料沒問題。
- 那我們會希望將內容能夠存下來，變成一個json檔案，傳給其他人。所以我們會使用open一個json file 後寫入成一個檔案。

```
import json
person_dict = {"name": "Bob", "age": 20, "sex": "F"}
person_json = json.dumps(person_dict)
print(person_json)
#with open('person.json', 'w') as json_file:
#    json.dump(person_dict, json_file)
```

39

Writing JSON to a file (中文處理)

- ensure_ascii=False：讓中文不被轉成 Unicode 編碼，例如 \u4f60\u597d。
- "w"：以「寫入模式」開啟檔案，若已存在同名檔會被覆寫。
- encoding="utf-8"：確保中文不亂碼。

```
import json
person_dict2 = {"姓名": "林小弘", "性別": "男", "學號": "B1231330", "專長": ["程式語言", "人工智慧"]}
with open('personc.json', 'w', encoding="utf-8") as json_file:
    json.dump(person_dict2, json_file, ensure_ascii=False)
```

40

讀取 JSON 檔案

- 我們也可以使用`json.load()` method 去讀取json檔案。
- 如果你有一個json檔為`person.json`，可以讀出`person.json`的內容
- 讀取 JSON 檔案的字串`json.load()` method，這個method 回傳一個 Dictionary

```
import json

with open("person.json", "r") as f:
    data = json.load(f) # 將 JSON 轉成 Python 字典
print(data)
#print(type(data))
```

41

讀取 JSON 檔案 & 參數

- 為了方便閱讀和debug JSON的資料，我們會將內容轉換為更易讀的格式。
- 這可以通過將附加參數 `indent` 和 `sort_keys` 傳遞給 `json.dumps()` 和 `json.dump()` 方法來完成。
- `Indent` 縮排的字元數(預設是None); `sort_keys` 進行升冪排序(預設是False)
- `indent=2`：讓輸出的 JSON 內容有縮排，易讀（美化格式）。
- 例如：

```
person_dict = {"name": "Bob", "age": 20, "sex": "M"}
print(json.dumps(data, indent = 2, sort_keys=True))
```

42

資料處理

- `extend()` or `append()`是把另一個可迭代物件（`iterable`，例如 `list`、`tuple`、`set`）中的元素逐一加入 `list` 中。
- `Update()`執行完程式後 `key` 相同JSON 檔的內容會被覆蓋掉（沒有保留原資料，只剩新資料）」，
- 若資料結構不同（例如一個是 `list`、一個是 `dict`），可轉換成統一格式後再合併。

特性	增加資料的方法	備註
資料是 dict	<code>dict.update()</code>	<code>key</code> 相同會覆蓋、 <code>key</code> 不同會新增
資料是 list	<code>list.extend()</code> or <code>list.append()</code>	在最後面加一或多筆資料
接 <code>open("file", "w")</code>	會覆蓋原資料	先讀舊資料再更新

43

資料處理 - 增加一筆資料

```
import json
new_data = {"name1": "Mary", "age1": 22, "sex1": "F"}

with open("person.json", "r", encoding="utf-8") as f1:
    dict1 = json.load(f1)
merged = dict1.copy()
merged.update(new_data)
with open("person_final.json", "w") as json_file:
    json.dump(merged, json_file, indent=2)
```

44

JSON參數

- 您還可以定義分隔符，默認值為(“,”,“:”)，表示使用逗號和空格分隔每個對象，使用冒號和空格分隔Key和value。
- 但是如果有自己定義的格式，也可以修改此分隔符號的默認值。
- 例如: 將, 改成點(.)， 將: 改成等號(=)。
- 就可以使用separator來進行設定。
- 例如:

```
print(json.dumps(person_dict, indent=4, separators=(". ", " = ")))
```

45

練習二

- 請以personc.json為基礎
- 開啟檔案後寫入你自己及另外一位小寶同學(資料隨便輸入)相關資訊，完成後儲存成personc_final.json檔案。
- 再讀取personc_final.json，以顯示出三位同學姓名、專長、性別及學號。

46

Q&A