

網頁程式設計 Web Programming

HTML 表單 HTML Forms

授課教師：林器弘

1

註解

- 註解很重要記得寫程式是給自己跟其他人看的，記得寫註解。
- `<!-- 註解掉的内容在這邊 -->`
- `Ctrl + /`

2

Input types

- Checkbox
- Email
- File
- Number
- Password
- Radio
- Image
- Date
- Time
- Hidden

3

Input Attributes

- Checked
- Value
- Placeholder
- Required

4

HTML input - required

- **【必要】** 格式用法很簡單，只需在input的標籤加入「required」。

```
[html]
<input name="text" id="text" type="text" required />
[/html]
```

5

HTML 表單 (form)

- HTML 表單 (form) 是用來讓使用者輸入資料，這些資料可以用來和使用者互動，例如表單內容填完後可以傳回遠端伺服器 (web server)，像常見的連絡表單。

```
<form action="formprocess.html" method="post">
```

```
<!-- ....表單 HTMLs .... -->
```

```
</form>
```

上面例子中，我們建立了一個表單，且宣告這個表單的資料會被用 http post 的方法送到 "formprocess.html" 這隻遠端程式。

6

HTML 表單 (form) 標籤 (tag)

- <form> 標籤是用來建立一個 HTML 表單，<form> 做為表單的容器，裡面還會有不同用途的其他標籤來建構出完整的表單內容。
- 再來介紹 <form> 標籤上面重要的屬性 (attribute)：
- action: 用來指定一個位址 (URL)，這個位址是告訴瀏覽器 (browser) 當 user 按送出表單後，要將表格的內容送去哪邊。

HTML 表單 (form) 標籤 (tag)

- method: 用來指定資料傳輸時用的 HTTP 協議，最常用的是 get 或 post：
 - get: 會將表單資料放在 form action 請求的位址 URL 網址參數列 (URL GET parameters) 上面送出。get 通常用在資料量較小或非敏感的資料，因為資料會被放在網址中直接傳出，容易被直接看到資料。
 - post: 會將表單資料放在 HTTP 傳輸封包 body 中送出。post 通常用在表單資料量比較大、有夾帶檔案上傳 (file upload) 或隱私性考量的資料。

HTML 表單 (form) 標籤 (tag)

- target: 用來指定瀏覽器要在何處顯示表單送出後伺服器回應的結果。值有這些選項：
 - _self: 顯示在表單所在的當前視窗
 - _blank: 顯示在新頁籤/視窗
 - _parent: 顯示在上一層的視窗 (用在例如如果表單是放在 <iframe> 中)
 - _top: 顯示在最頂層的視窗
- autocomplete: 指示這個表單中的欄位是否啟用瀏覽器自動完成機制。值有這些選項：
 - off: 否
 - on: 是，預設值

HTML 表單 (form) 標籤 (tag)

```
<form action="formprocess.html" method="post">

  <p>Name:<input type="text" name="name" value="你的名字"></p>
  <p>Email:<input name="species" type="text"></p>
  <p>Comments: </p>
  <p><textarea name="comments" rows="5" cols="20">你的留言</textarea></p>

  <p>請問你最有興趣學習的技術:</p>
  <p><input type="radio" name="interest" value="html"> HTML</p>
  <p><input type="radio" name="interest" value="css"> CSS</p>
  <p><input type="radio" name="interest" value="js"> JavaScript</p>

  <p><input type="submit" value="送出資料"></p>

</form>
```

練習四

- 請設計一個註冊網頁
- 其中內容需要包含
 - 姓名、性別、單位、e-mail（帳號）、密碼、聯絡電話。
- 送出後顯示粗體文字：註冊完成，靜待開通權限。
- 其中：
 - 性別使用select
 - 單位使用datalist: 研發部、行政部、管理部

姓名:

性別:

單位:

e-mail:

密碼:

聯絡電話:

11

HTML 表單 (form) - pattern

正規式	說明及範例	比對不成立之字串
/a/	含字母 "a" 的字串，例如 "ab", "bac", "cba"	"xyz"
/a./	含字母 "a" 以及其後任一個字元的字串，例如 "ab", "bac" (若要比對.，請使用 \.)	"a", "ba"
/^xy/	以 "xy" 開始的字串，例如 "xyz", "xyab" (若要比對 ^，請使用 \^)	"axy", "bxy"
/xy\$/	以 "xy" 結尾的字串，例如 "axy", "abxy" (若要比對 \$，請使用 \\$)	"xya", "xyb"
[13579]	包含 "1" 或 "3" 或 "5" 或 "7" 或 "9" 的字串，例如: "a3b", "1xy"	"y2k"
[0-9]	含數字之字串	不含數字之字串
[a-z0-9]	含數字或小寫字母之字串	不含數字及小寫字母之字串
[a-zA-Z0-9]	含數字或字母之字串	不含數字及字母之字串
b[aeiou]t	"bat", "bet", "bit", "bot", "but"	"bxt", "bzt"
[^0-9]	不含數字之字串 (若要比對 ^，請使用 \^)	含數字之字串
[^aeiouAEIOU]	不含母音之字串 (若要比對 ^，請使用 \^)	含母音之字串
[^^\]	不含 "^" 之字串，例如 "xyz", "abc"	"xy^", "a^bc"

"^" 在兩條斜線中，代表一個字串的開始位置，因此 /^xy/ 代表以 "xy" 開始的字串。同理，"\$" 在兩條斜線中，代表一個字串的結束位置，因此 /xy\$/ 代表以 "xy" 結束的字串。但是如果將 "^" 放在兩個方括弧中，就代表「否定」，因此 [^aeiouAEIOU] 代表不含母音之字串。

另外，若要避掉特殊字元的特殊意義，就要在此字元前加上 "\"，例如上表中的最後一列，"^" 在方括弧裡面是代表「否定」，因此若要在方括弧裡面比對 "^"，就要使用 "\\^"，所以「不含 "^" 之字串」的通用式就是 "[^\\^]"。

12

HTML 表單 (form) - pattern

- 有些正規式會常被用到，因此已被定義為特定字元，以簡化整體通用式，這些字元可列表說明如下：

正規表示法的特定字元	說明	等效的正規表示法
\d	數字	[0-9]
\D	非數字	[^0-9]
\w	數字、字母、底線	[a-zA-Z0-9_]
\W	非 \w	[^a-zA-Z0-9_]
\s	空白字元	[\r\t\n\f]
\S	非空白字元	[^\r\t\n\f]

13

HTML 表單 (form) - pattern

- pattern:
- 這個屬性使用正規表示法進行驗證，可以用正規表示法來驗證表單內容格式，當驗證錯誤會給出提醒。

```
<form>  
<label>密碼:<input type="password" name="pwd"  
pattern="[a-zA-Z0-9]{7,}" /></label><input type="submit"  
name="SB" value="送出" />  
</form>
```

- 範例的規範為限定輸入大寫或小寫英文以及數字，字元最少要7個以上。

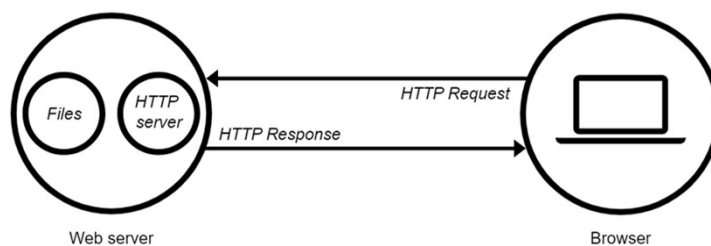
14

Install and Run Web Server

15

何謂網路伺服器？

- 以最基本的層面來說，如果瀏覽器需要網路伺服器所託管的檔案，它就需要透過 HTTP 發送對該檔案的請求。如果請求已經傳送到正確的（硬體）網路伺服器，那 HTTP（軟體）伺服器就會接受請求、找出所請求的文件（假若不是接著回傳 404 頁面）、再透過 HTTP 回傳給瀏覽器。



16

何謂網路伺服器？

- 在網路伺服器裡面，HTTP 伺服器負責處理和回答傳入的請求。
 - HTTP 伺服器接收請求後，會先檢查請求的 URL 是否匹配現有文件。
 - 如果匹配，網路伺服器會把檔案內容回傳給瀏覽器。不然，應用伺服器會建立需要的檔案。
 - 如果都沒有用的話，網路伺服器會回傳錯誤訊息給瀏覽器，最常見的就是「404 Not Found」。（這個錯誤很常見，所以許多網頁設計師花了相當多的心力設計 404 錯誤頁面。）

Not Found

We're sorry, we couldn't find what you were looking for.
You can try a search or start over on the home page.
If you were following a documentation link, you can sign in and create the page. Otherwise, please fix a bug. Thank!

[Sign in with Persona](#)

Turn on by default from all the content



TikTok

Log in

4 😊 4

Couldn't find this page

Check out more trending videos on TikTok

[Watch now](#)

17

基本概念

Server



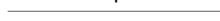
User



HTTP request



HTTP response



18

pip install flask

- python -m pip install

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\111_HTML\20230310> python -m pip install flask
Requirement already satisfied: flask in d:\python36\lib\site-packages (2.0.3)
Requirement already satisfied: Jinja2>=3.0 in d:\python36\lib\site-packages (from flask) (3.0.3)
Requirement already satisfied: Werkzeug>=2.0 in d:\python36\lib\site-packages (from flask) (2.0.3)
Requirement already satisfied: click>=7.1.2 in d:\python36\lib\site-packages (from flask) (8.0.4)
Requirement already satisfied: itsdangerous>=2.0 in d:\python36\lib\site-packages (from flask) (2.0.1)
Requirement already satisfied: colorama in d:\python36\lib\site-packages (from click>=7.1.2->flask) (0.4.4)
Requirement already satisfied: importlib-metadata in d:\python36\lib\site-packages (from click>=7.1.2->flask) (4.8.3)
Requirement already satisfied: MarkupSafe>=2.0 in d:\python36\lib\site-packages (from Jinja2>=3.0->flask) (2.0.1)
Requirement already satisfied: dataclasses in d:\python36\lib\site-packages (from Werkzeug>=2.0->flask) (0.8)
Requirement already satisfied: typing-extensions>=3.6.4 in d:\python36\lib\site-packages (from importlib-metadata->click>=7.1.2->flask) (4.1.1)
Requirement already satisfied: zipp>=0.5 in d:\python36\lib\site-packages (from importlib-metadata->click>=7.1.2->flask) (3.6.0)
PS D:\111_HTML\20230310>
```

19

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python 3.11.7 (tags/v3.11.7:7a2a02f, Dec 4 2023, 19:24:40) [MSI v.1017 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
KeyboardInterrupt
>>> exit()
PS D:\111_HTML\20230310> python -m pip install flask
Collecting flask
  Obtaining dependency information for flask from https://files.pythonhosted.org/packages/61/80/fefda13a9306f7c93af113dd9638c75138c42a0994bdcfac078b96/flask-3.0.3-py3-none-any.whl.metadata
  Downloading flask-3.0.3-py3-none-any.whl.metadata (3.2 kB)
Collecting Werkzeug>=2.0 (from flask)
  Obtaining dependency information for Werkzeug from https://files.pythonhosted.org/packages/4b/94/997bf7c2bf2dc3f4956c6d840959fef535c184096cfd26d3a6705b/werkzeug-3.0.4-py3-none-any.whl.metadata
  Downloading werkzeug-3.0.4-py3-none-any.whl.metadata (2.7 kB)
Collecting Jinja2>=3.1.2 (from flask)
  Obtaining dependency information for Jinja2 from https://files.pythonhosted.org/packages/31/80/3a683c3f461f6ec263ebf3a14171bb85198238b346a0a0540c36/jinja2-3.1.4-py3-none-any.whl.metadata
  Downloading Jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.1.2 (from flask)
  Obtaining dependency information for itsdangerous from https://files.pythonhosted.org/packages/94/96/92447266d6d9282a776cf8b28ba9b07c0596252a6f0e408583f41/itsdangerous-2.2.0-py3-none-any.whl.metadata
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting click>=8.1.3 (from flask)
  Obtaining dependency information for click from https://files.pythonhosted.org/packages/0b/2a/053fa8ebf2cf671330affc7ca780c04f16b71827773b0311a3229/click-8.1.7-py3-none-any.whl.metadata
  Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting blinker>=1.6.2 (from flask)
  Obtaining dependency information for blinker from https://files.pythonhosted.org/packages/bb/2a/1816edf1116a2f7139936a821765c6251eadb30a52bde14640b/blinker-1.8.2-py3-none-any.whl.metadata
  Downloading blinker-1.8.2-py3-none-any.whl.metadata (1.6 kB)
Collecting colorama (from click>=8.1.3->flask)
  Obtaining dependency information for colorama from https://files.pythonhosted.org/packages/d1/96/3965e084c3042e047c6a3e0ed1a03a750876a09aa3a150d8ac56c221/colorama-0.4.6-py2.py3-none-any.whl.metadata
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (2.7 kB)
Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->flask)
  Obtaining dependency information for MarkupSafe from https://files.pythonhosted.org/packages/b7/a2/c78a848ec584b3445a4961554c7e086a82b0b844e75410b57087cc/MarkupSafe-2.1.5-cp11-cp11-win_msd64.whl.metadata
  Downloading MarkupSafe-2.1.5-cp11-cp11-win_msd64.whl.metadata (1.1 kB)
Downloading flask-3.0.3-py3-none-any.whl (101 kB)
101.7/101.7 kB 977.4 kB/s eta 0:00:00
Downloading blinker-1.8.2-py3-none-any.whl (10.5 kB)
10.5/10.5 kB 977.4 kB/s eta 0:00:00
Downloading click-8.1.7-py3-none-any.whl (97 kB)
97.0/97.0 kB 2.5 MB/s eta 0:00:00
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
16.0/16.0 kB 2.7 MB/s eta 0:00:00
Downloading Jinja2-3.1.4-py3-none-any.whl (133 kB)
133.0/133.0 kB 2.7 MB/s eta 0:00:00
Downloading werkzeug-3.0.4-py3-none-any.whl (222 kB)
222.0/222.0 kB 2.7 MB/s eta 0:00:00
Downloading MarkupSafe-2.1.5-cp11-cp11-win_msd64.whl (22 kB)
22.0/22.0 kB 2.8 MB/s eta 0:00:00
Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
25.0/25.0 kB 2.8 MB/s eta 0:00:00
Installing collected packages: MarkupSafe, itsdangerous, colorama, blinker, Werkzeug, Jinja2, click, flask
Successfully installed Jinja2-3.1.4 MarkupSafe-2.1.5 Werkzeug-3.0.4 blinker-1.8.2 click-8.1.7 colorama-0.4.6 flask-3.0.3 itsdangerous-2.2.0
```

20

Hello flask – App_hello.py

```
from flask import Flask

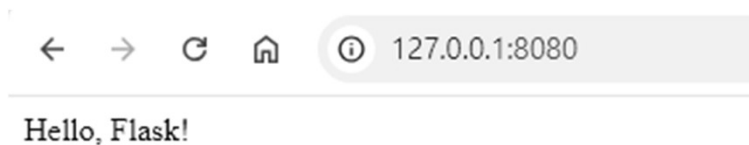
app = Flask(__name__)
@app.route("/")
def index():
    return "Hello, Flask!"

if __name__ == '__main__':
    app.run(host='0.0.0.0',port='8080',debug=True)
```

21

觀察網頁結果

- 接下來開啟瀏覽器，輸入：http://127.0.0.1:8080/



```
Press CTRL+C to quit
127.0.0.1 - - [17/Sep/2024 14:50:01] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [17/Sep/2024 14:50:01] "GET /favicon.ico HTTP/1.1" 404 -
```

22

專案結構

- reg_project/
 - └─ app.py
 - └─ templates/
 - └─ register.html
 - └─ result.html

23

Server - App.py

- render_template是flask的內建函數，字面上的意思大概是渲染模板，因此它需要一個『模板』來提供渲染的格式。
- 直觀來看，當return render_template('html文件')的時候，flask會先到專案資料夾『templates』去尋找相對應的html文件，因此，你需要先做的就是在專案底下建置一個資料夾，並且命名為『templates』

24

Import package - test.py

```
from flask import Flask, render_template

app = Flask(__name__)
@app.route('/')
def index():
    return render_template('test.html')

if __name__ == '__main__':
    app.run(host='0.0.0.0',port='8080',debug=True)
```

25

import

- `render_template` : 載入 `templates/` 資料夾裡的 HTML 檔並渲染
- `request` : 取得前端送過來的資料 (例如 `request.form`、`request.args`)
- `redirect` : 讓使用者導向到另一個路由
- `url_for` : 產生對應路由的 URL

26

Server – App0.py (1/2)

```
from flask import Flask, render_template, request, redirect, url_for
app = Flask(__name__)
@app.route("/")
def index():
    return redirect(url_for("register"))
@app.route("/register", methods=["GET", "POST"])
def register():
    if request.method == "POST":
        <候補資料>
        return redirect(url_for(<候補資料>))
    return render_template("reg0.html")
```

27

Server – App0.py(2/2)

```
@app.route("/result")
def result():
    <候補資料>
    return render_template(<候補資料>)

if __name__ == '__main__':
    app.run(host='0.0.0.0', port='8080', debug=True)
```

28

reg0.html

- 在templates資料夾內新增一個html文件並命名為reg0.html and res0.html，內容如下：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>註冊</title>
</head>
<body>
  <h1>註冊</h1>
  <form method="post" action="{{ url_for('register') }}">
    <p>
      姓名: <input type="text" name="name">
    </p>
    <p>
      Email: <input type="email" name="email">
    </p>
    <p>
      密碼: <input type="password" name="password">
    </p>
    <p>
      <button type="submit">送出</button>
    </p>
  </form>
</body>
</html>
```

29

res0.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>註冊結果</title>
</head>
<body>
  <h1>註冊成功！</h1>
  <p>姓名：{{ name }}</p>
  <p>Email：{{ email }}</p>
  <p><a href="{{ url_for('register') }}">返回註冊頁</a></p>
</body>
</html>
```

30

Server – App0.py (1/2)

```
from flask import Flask, render_template, request, redirect, url_for
app = Flask(__name__)
@app.route("/")
def index():
    return redirect(url_for("register"))
@app.route("/register", methods=["GET", "POST"])
def register():
    if request.method == "POST":
        form = {
            "name": request.form.get("name", ""),
            "email": request.form.get("email", ""),
            "password": request.form.get("password", ""),
        }
        return redirect(url_for("result", name=form["name"], email=form["email"]))
    return render_template("reg0.html")
```

31

Server – App0.py(2/2)

```
@app.route("/result")
def result():
    name = request.args.get("name", "")
    email = request.args.get("email", "")
    return render_template("res0.html", name=name, email=email)

if __name__ == '__main__':
    app.run(host='0.0.0.0', port='8080', debug=True)
```

32

練習五

- 請以練習四為基礎的註冊頁面
- 送出資料後在網頁上顯示出您所Key的資訊
- 並限制電話需要十碼

註冊

姓名：

Email：

單位：

連絡電話：

建議：

註冊成功！

姓名：123

Email：123@123.123.tw

單位：行政部

連絡電話：1231231231

建議：123

[回到註冊頁](#)

33

reg.html

- 在templates資料夾內新增一個html文件並命名為reg.html，內容如下：

```
<form action="" method="post">
<p>
  姓名：
  <textarea name="suggestion" rows="4" cols="40"></textarea>
  <button type="submit">送出</button>
```

34

Server – App.py

#上課練習五 (Server - App.py)

35

Q&A

36

36