



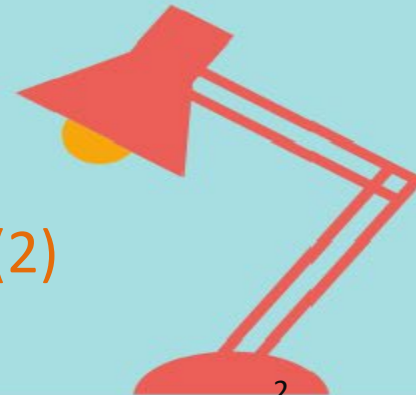
111-2進階程式設計課程(14)

Advanced Computer Programming

亞大資工系

課程大綱

- W1-課程介紹/Introduction
- W2-Python libraries
- W3-BeautifulSoup(1)
- W4-BeautifulSoup(2)
- W5-
- W6-Scrapy(1)
- W7-Scrapy(2)
- W8-Storing Data
- W9-Midterm project
- W10-Web & HTTP
- W11-Flask
- W12-Flask Routes
- W13-Jinja template
- W14-Flask-form
- W15-Flask-mail
- W16-REST API
- W17-Project development(2)
- W18-Final presentation





python



Flask

web development,
one drop at a time



作業3

- 選擇一個Bootstrap Personal Templates
- 將網頁上的資料及照片換成你個人的資料
- 參考老師的教學步驟，重新上傳至pythonanywhere網站
- 繳交pythonanywhere網址



(14) Flask-WTF

- Creating Forms
- Validating Forms
- CSRF
 - Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.



Example 4-1. hello.py: Flask-WTF configuration

```
app = Flask(__name__)  
app.config['SECRET_KEY'] = 'hard to  
guess string'
```



Example 4-2. hello.py: form class definition

```
from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField
from wtforms.validators import DataRequired

class NameForm(FlaskForm):
    name = StringField('What is your name?', validators=[DataRequired()])
    submit = SubmitField('Submit')
```

Table 4-1. WTForms standard HTML fields

Field type	Description	Field type	Description
BooleanField	Checkbox with True and False values	FloatField	Text field that accepts a floating-point value
DateField	Text field that accepts a <code>datetime.date</code> value in a given format	FormField	Form embedded as a field in a container form
DateTimeField	Text field that accepts a <code>datetime.datetime</code> value in a given format	IntegerField	Text field that accepts an integer value
DecimalField	Text field that accepts a <code>decimal.Decimal</code> value	PasswordField	Password text field
FileField	File upload field	RadioField	List of radio buttons
HiddenField	Hidden text field	SelectField	Drop-down list of choices
MultipleFileField	Multiple file upload field	SelectMultipleField	Drop-down list of choices with multiple selection
FieldList	List of fields of a given type	SubmitField	Form submission button
		StringField	Text field



Table 4-2. WTForms validators

Validator	Description
<code>DataRequired</code>	Validates that the field contains data after type conversion
<code>Email</code>	Validates an email address
<code>EqualTo</code>	Compares the values of two fields; useful when requesting a password to be entered twice for confirmation
<code>InputRequired</code>	Validates that the field contains data before type conversion
<code>IPAddress</code>	Validates an IPv4 network address
<code>Length</code>	Validates the length of the string entered
<code>MacAddress</code>	Validates a MAC address
<code>NumberRange</code>	Validates that the value entered is within a numeric range
<code>Optional</code>	Allows empty input in the field, skipping additional validators
<code>Regexp</code>	Validates the input against a regular expression
<code>URL</code>	Validates a URL
<code>UUID</code>	Validates a UUID
<code>AnyOf</code>	Validates that the input is one of a list of possible values
<code>NoneOf</code>	Validates that the input is none of a list of possible values



Example 4-3. templates/index.html

- `{% extends "base.html" %}`
- `{% import "bootstrap/wtf.html" as wtf %}`
- `{% block title %}Flasky{% endblock %}`
- `{% block page_content %}`
- `<div class="page-header">`
- `<h1>Hello, {% if name %}{{ name }}{% else %}Stranger{% endif %}!</h1>`
- `</div>`
- `{{ wtf.quick_form(form) }}`
- `{% endblock %}`

Example 4-4. hello.py: handle a web form with GET and POST request methods

```
@app.route('/', methods=['GET', 'POST'])
def index():
    name = None
    form = NameForm()
    if form.validate_on_submit():
        name = form.name.data
        form.name.data = ''
    return render_template('index.html', form=form, name=name)
```

Example 4-5. hello.py: redirects and user sessions

```
from flask import Flask, render_template, session, redirect, url_for

@app.route('/', methods=['GET', 'POST'])
def index():
    form = NameForm()
    if form.validate_on_submit():
        session['name'] = form.name.data
        return redirect(url_for('index'))
    return render_template('index.html', form=form, name=session.get('name'))
```

Example 4-6. hello.py: flashed messages

```
from flask import Flask, render_template, session, redirect, url_for, flash

@app.route('/', methods=['GET', 'POST'])
def index():
    form = NameForm()
    if form.validate_on_submit():
        old_name = session.get('name')
        if old_name is not None and old_name != form.name.data:
            flash('Looks like you have changed your name!')
        session['name'] = form.name.data
        return redirect(url_for('index'))
    return render_template('index.html',
        form = form, name = session.get('name'))
```

Example 4-7. templates/base.html: rendering of flashed messages

```
{% block content %}
<div class="container">
    {% for message in get_flashed_messages() %}
    <div class="alert alert-warning">
        <button type="button" class="close" data-
dismiss="alert">&times;</button>
        {{ message }}
    </div>
    {% endfor %}

    {% block page_content %}{% endblock %}
</div>
{% endblock %}
```

Thanks!

Q&A

